Binary Matrix Factorization via Collaborative Neurodynamic Optimization

Hongzong Li[a], Jun Wang[a,b,*], Nian Zhang[c], Wei Zhang[d]

[a] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
[b] School of Data Science, City University of Hong Kong, Kowloon, Hong Kong
[c] Department of Electrical & Computer Engineering, University of the District of Columbia, Washington, DC, USA
[d] Chongqing Engineering Research Center of Internet of Things and Intelligent Control Technology, Chongqing Three Gorges University, Chongqing, China

Abstract

Binary matrix factorization is an important tool for dimension reduction for high-dimensional datasets with binary attributes and has been successfully applied in numerous areas. This paper presents a collaborative neurodynamic optimization approach to binary matrix factorization based on the original combinatorial optimization problem formulation and quadratic unconstrained binary optimization problem reformulations. The proposed approach employs multiple discrete Hopfield networks operating concurrently in search of local optima. In addition, a particle swarm optimization rule is used to reinitialize neuronal states iteratively to escape from local minima toward better ones. Experimental results on eight benchmark datasets are elaborated to demonstrate the superior performance of the proposed approach against six baseline algorithms in terms of factorization error. Additionally, the viability of the proposed approach is demonstrated for pattern discovery on three datasets.

Keywords: Binary matrix factorization; collaborative neurodynamic optimization; discrete Hopfield network; quadratic unconstrained binary optimization; pattern discovery.

1. Introduction

Binary matrix factorization (BMF) is an essential tool for identifying discrete patterns within binary data. It approximates a given binary matrix $V \in \mathbb{R}^{n\times m}$ by determining two factor matrices $X \in \mathbb{R}^{n\times r}$ and $Y \in \mathbb{R}^{r\times m}$, where $0 < r \ll \min(n, m)$, with the objective of minimizing the Frobenius loss $||XY - V||_F^2$. It has various applications, including graph partitioning (Chandran et al. (2017)), low-density parity check coding (Ravanbakhsh et al. (2016)), LED-display optimization (Kumar et al. (2019)), association rule mining (Koyutürk & Grama (2003)), structure identification biclustering for gene expression (Zhang et al. (2007); Zhang et al. (2010)), pattern discovery (Shen et al. (2009a)), digits reconstruction (Meeds et al. (2006)), discrete-attribute data mining (Koyuturk et al. (2005); Koyutürk et al. (2006)), market data clustering (Li (2005)), document clustering (Zhang et al. (2007)), role-based access control (Lu et al. (2008, 2014)), and etc.

The challenge in BMF lies in the combinatorial nature of the optimization problem. In the view that the BMF problem is NP-hard (Gillis & Vavasis (2018); Dan et al. (2018)), approximation and heuristic methods are widely used. Approximation methods allow X and Y to take real values and then approximate solutions to the binary domain using certain predefined rules; e.g., Zhang et al. (2007); Slawski et al. (2013); Diop et al. (2017). Existing heuristic methods include the Proximus algorithm (Koyutürk et al. (2002); Koyutürk & Grama (2003)), the association rule-mining algorithm (Miettinen et al. (2008)), the consensus algorithm (Fu et al. (2010)), the clustering-based algorithm (Jiang et al. (2014)), the divide-and-conquer algorithm (Beckerleg & Thompson (2020)), and etc. Meta-heuristic methods include the genetic algorithm (Snášel et al. (2008)), etc.

In his seminal papers (Hopfield (1982); Hopfield & Tank (1986)), John Hopfield heralds that the networks of simple and similar neurons collectively can serve as powerful computation models (known as Hopfield networks). Over the past few decades, various neurodynamic optimization models have emerged to solve diverse optimization problems, such as nonconvex and global optimization problems (e.g., Che & Wang (2019); Wei et al. (2024); Jin et al. (2024)), nonsmooth pseudoconvex optimization (e.g., Liu et al. (2022)) combinatorial optimization problems (e.g., Hopfield & Tank (1985); Che & Wang (2019)), and other related problems (Ju et al. (2023, 2024a,b)).

It is recognized that a single neurodynamic model en-

*This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region of China under Grants 11202019, and 11203721.

*Corresponding author

Email addresses: hongzli2-c@my.cityu.edu.hk (Hongzong Li), jwang.cs@cityu.edu.hk (Jun Wang), nzhang@udc.edu (Nian Zhang), weizhang@sanxiau.edu.cn (Wei Zhang)

counters difficulties in efficiently tackling combinatorial optimization problems with binary variables, as a single gradient-driven neurodynamic model may lead to local optimal solutions. In recent years, the collaborative neurodynamic optimization (CNO) approach has been developed as a hybrid intelligence framework. It combines neurodynamic optimization with evolutionary optimization for solving various complex optimization problems. CNO employs a population of individual neurodynamic optimization models for exploring local optimal solutions and incorporates a meta-heuristic rule (e.g., particle swarm optimization), for updating initial neuronal states to escape from local minima and facilitating the exploration of global optima. A mutation operator may be used to maintain diversity in initial neuronal states for preventing premature convergence. It is proven in Yan et al. (2017) that collaborative neurodynamic approaches are almost surely convergent to the global optimal solutions of the optimization problems (Yan et al. (2014); Che & Wang (2019); Che & Wang (2021)). In the framework of collaborative neurodynamic optimization, several approaches are developed for solving nonconvex and global optimization problems (e.g., Yan et al. (2014); Che & Wang (2019); Xia et al. (2024)), distributed optimization (e.g., Jia et al. (2024); Huang et al. (2024)), distributed minimax optimization (e.g., Xia et al. (2023)), and combinatorial optimization problems (e.g., Che & Wang (2019); Che & Wang (2021)). CNO approaches are used as computationally intelligent optimizers in various applications such as nonnegative matrix factorization (Che & Wang (2018)), bicriteria sparse nonnegative matrix factorization (Che et al. (2023)), Boolean matrix factorization (Li et al. (2022)), financial portfolio selection (Leung et al. (2022); Leung & Wang (2022)), and sparse signal reconstruction (Che et al. (2022)).

In this paper, we propose a neurodynamic-driven algorithm for BMF in the framework of CNO (CNO-BMF). The proposed algorithm consists of a phase with DHNm's updated synchronously and another phase with DHNs updated synchronously in batches. It leverages multiple discrete Hopfield networks and a particle swarm optimization update rule to reinitialize discrete Hopfield networks for escaping from local optima and moving toward global optimal solutions. We demonstrate its superior performance against six prevailing baselines in terms of factorization loss. In addition, we also apply the proposed approach for pattern discovery on three datasets.

The contributions of this work are summarized as follows.

i. We propose the CNO-BMF algorithm utilizes efficient exploration capability of discrete Hopfield network with momentum term in scattered searches and the gradient-free updating feature of a particle swarm optimization rule to reposition the neuronal searches escaping from local minima.

ii. We experimentally demonstrate that the CNO-BMF algorithm statistically outperforms six prevailing baselines in terms of factorization loss.

iii. We experimentally illustrate the effectiveness of the CNO-BMF algorithm applied in pattern discovery.

The remainder of this paper is arranged as follows. The preliminaries on discrete Hopfield network and collaborative neurodynamic optimization are provided in Section 2. The problem formulation is stated in Section 3. The details of the CNO-BMF algorithm are presented in Section 4. Experimental results on eight datasets are reported in Section 5. A specific application of BMF on pattern discovery is provided in Section 6. The paper is concluded in Section 7.

2. Preliminaries

2.1. Discrete Hopfield Network

The discrete Hopfield network (DHN) stands as a classic recurrent neural network distinguished by its binary or bipolar states and activation function operating in discrete time as follows (Hopfield (1982)):

$$\begin{cases} u(t+1) = Wx(t) + \theta, \\ x(t) = g(u(t)), \end{cases} \tag{1}$$

where $u \in \Re^n$ is the net-input vector, $x \in \Re^n$ is the state vector, $W \in \Re^{n\times n}$ is the connection weight matrix, $\theta \in \Re^n$ is the threshold vector, and $\sigma(\cdot)$ is a vector-valued discontinuous activation function defined element-wisely as follows:

$$g(u_i) = \begin{cases} 0, & u_i(t) \le 0, \\ 1, & \text{otherwise}. \end{cases}$$

It is demonstrated in (Hopfield (1982)) that DHN in (1) is globally stable at an equilibrium $\bar{x}$ (i.e., $\lim_{t\to\infty} x(t) = \bar{x}$) provided that the connection weight matrix is symmetric (i.e., $W = W^T$), the main diagonal elements of W are zero (i.e., $w_{ii} = 0$, $\forall i$), and the activation is carried out asynchronously. Furthermore, it is demonstrated in Hopfield (1982) that the DHN globally converges to a local minimum of the following combinatorial optimization problem:

$$\begin{aligned} \min_{x} \ & -\frac{1}{2}x^T W x - \theta^T x, \\ \text{s.t. } & x \in \{0,1\}^n. \end{aligned} \tag{2}$$

An equilibrium point $\bar{x}$ of the discrete Hopfield network is a local optimum for the optimization problem above. It is noteworthy that the right-hand side of eqn. (1) is the positive gradient of the objective function to be maximized or the negative gradient of the objective function to be minimized. In essence, the neurodynamics of the DHN form a discrete gradient flow, moving among the vertices of the unit hypercube coordinate-wisely.

Given the binary nature of state variable $x_i \in \{0,1\}$, it follows that $x_i^2 = x_i$ for $i = 1, 2, \dots, n$. Consequently, the diagonal elements of the weight matrix in the quadratic term of (1) can always be set to zeros by introducing an equivalent linear term $\mathrm{diag}(w_{11}, \dots, w_{nn})x$.

The DHN's solution quality depends on the sequence of activations. For certain W possessing special properties, synchronous activation in batches may mitigate the sequence dependence of solution quality. Various methods are developed for synchronous activation of neuronal states in batches; e.g., Cernuschi-Frías (1989); Likas & Stafylopatis (1996); Lee (1999); Muñoz-Pérez et al. (2011). For example, the DHN is still convergent to a local minimum if the neurons without any direct connections are activated synchronously in batches (Muñoz-Pérez et al. (2011)).

A DHN with a momentum term (DHNm) is introduced (Takefuji & Lee (1989)) with the following neurodynamic equation:

$$\begin{cases} u(t) = u(t-1) + Wx(t) - \theta, \\ x(t+1) = \sigma(u(t)). \end{cases} \tag{3}$$

DHNm (3) takes historical effects into account and enriches its dynamic behaviors by including the momentum term $u(t-1)$. It has been demonstrated that the synchronously activated neuronal states of DHNm (3) are convergent to a local optimum of (2) (Takefuji & Lee (1991); Galán-Marín & Muñoz-Pérez (2001)).

2.2. Collaborative Neurodynamic Optimization

In the existing CNO paradigms, projection neural networks (e.g., Wang et al. (2020)) and DHNs (e.g., Wang et al. (2021)) are often used for local searches. A particle swarm optimization rule is used in almost all of the CNO algorithms to reposition the initial states of the neurodynamic models. Among the various particle swarm optimization rules, the von Neumann topology stands out as an effective and well-studied variant (Kennedy & Mendes (2002)). In this topology, particles are organized in a grid-like structure, forming a lattice of interconnected neighborhoods. Let p_i^* denote the best position found by the i-th particle individually, p_i denote the position vector of the i-th particle, l_i^* denotes the best neighbor of the i-th particle on all four sides of the two-dimensional lattice, and N denote the number of particles. The velocity v_i and the position p_i, for $i = 1, 2, \dots, N$, are updated as follows:

$$\begin{cases} v_i(t+1) = c_0 v_i(t) + c_1 r_1 (p_i^*(t) - p_i(t)) + \\ \qquad\qquad c_2 r_2 (l_i^*(t) - p_i(t)), \\ \text{if } (r_3 < S(v_{id}(t))), \text{then } p_{id}(t) = 1, \text{else } p_{id}(t) = 0, \end{cases} \tag{4}$$

where c_0 is an inertia parameter, c_1, c_2 are two acceleration constants, $r_1, r_2, r_3 \in [0,1]$ are three random numbers, and $S(\cdot)$ is a sigmoid limiting transformation.

The diversity of global search is non-negligible in global and combinatorial optimization in the presence of convexity in objective functions or solution spaces. A simple diversity measure is defined as:

$$\delta(x) = \frac{1}{Nn} \sum_{i=1}^{N} \|p_i - p^*\|_2, \tag{5}$$

where n is the dimension of solutions, and p^* is the best solution among the N solutions.

In the literature, many mutation operators are used to ensure solution diversity. In particular, the following bit-flip mutation operation is defined in Zhang et al. (2014): if $\delta(x) < \delta_{\min}$, then

$$x_j = \begin{cases} \neg x_j & \text{if } \xi_j \le \rho, \\ x_j & \text{otherwise }, \end{cases} \tag{6}$$

where $\delta_{\min}$ is a threshold, $\bar{x}_j$ is the negation of x_j, ξ_j is a randomly generated number in the range of $[0,1]$, ρ is a mutation probability.

3. Problem Formulations

Consider the following binary matrix factorization problem:

$$\begin{aligned} \min_{X,Y} \quad & f(X,Y) := ||XY - V||_F^2, \\ \text{s.t.} \quad & X \in \{0,1\}^{n \times r},\ Y \in \{0,1\}^{r \times m}, \end{aligned} \tag{7}$$

where $|| \cdot ||_F$ is the Frobenius norm, $V \in \{0,1\}^{n \times m}$ is a given matrix of binary data, X and Y are unknown matrices of binary factors.

Let $\tilde{x}_i \in \{0,1\}^r$ denote the i-th row of X and $y_j \in \{0,1\}^r$ denote the j-th column of Y for $i = 1, 2, \dots, n; j = 1, 2, \dots, m$.

$$\begin{aligned} & ||XY - V||_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} (\tilde{x}_i y_j - v_{ij})^2 = \\ & \sum_{i=1}^{n} \sum_{j=1}^{m} ((\tilde{x}_i y_j)^2 - 2 v_{ij} \tilde{x}_i y_j + v_{ij}^2) = \\ & \sum_{i=1}^{n} \sum_{j=1}^{m} (y_j^T \tilde{x}_i^T \tilde{x}_i y_j - 2 v_{ij} \tilde{x}_i y_j + v_{ij}^2). \end{aligned} \tag{8}$$

In view that $x_{ik}^2 = x_{ik}$ and $y_{kj}^2 = y_{kj}$, the fourth-degree monomial in (8)

$$\begin{aligned} & y_j^T \tilde{x}_i^T \tilde{x}_i y_j = \sum_{k=1}^{r} \sum_{l=1}^{r} x_{ik} x_{il} y_{kj} y_{lj} = \\ & \sum_{k=1}^{r} \sum_{l \ne k} x_{ik} x_{il} y_{kj} y_{lj} + \sum_{k=1}^{r} x_{ik} y_{kj}. \end{aligned} \tag{9}$$

As a result,

$$\begin{aligned} & ||XY - V||_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \Big\{ \sum_{k=1}^{r} \big[\sum_{l \ne k} x_{ik} x_{il} y_{kj} y_{lj} + \\ & (1 - 2 v_{ij}) x_{ik} y_{kj} \big] + v_{ij}^2 \Big\}. \end{aligned} \tag{10}$$

4. Algorithm Description

To facilitate BMF, the problem in (10) is treated as two quadratic binary problems, one in X with fixed Y, and the other in Y with fixed X.

The partial derivatives of the objective function in $f(X,Y)$ with respect to the elements x_{ij} and y_{jk} are derived as follows, for $i = 1, 2, \ldots, n; j = 1, 2, \ldots, r; k = 1, 2, \ldots, m$:

$$\frac{\partial ||XY - V||_F^2}{\partial x_{ij}} = \sum_{k=1}^{m} \left(\sum_{l \neq j} 2x_{il}y_{lk} + (1 - 2v_{ik}) \right) y_{jk}, \tag{11}$$

$$\frac{\partial ||XY - V||_F^2}{\partial y_{jk}} = \sum_{i=1}^{n} \left(\sum_{l \neq j} 2x_{il}y_{lk} + (1 - 2v_{ik}) \right) x_{ij}. \tag{12}$$

In DHNs, X and Y denote matrix-value neuronal states. Based on the derived partial derivatives in (11) and (12), the activation functions of DHNs for updating X and Y in BMF are written as follows, respectively:

$$\begin{aligned} U_X(t+1) &= -\nabla_X ||X(t)Y(t) - V||_F^2, \\ &= \left[-\sum_{k=1}^{m} \left(\sum_{l \neq j} 2x_{il}y_{lk} + (1 - 2v_{ik}) \right) y_{jk} \right]_{ij}, \\ X(t) &= g(U_X(t)), \end{aligned} \tag{13}$$

$$\begin{aligned} U_Y(t+1) &= -\nabla_Y ||X(t)Y(t) - V||_F^2, \\ &= \left[-\sum_{i=1}^{n} \left(\sum_{l \neq j} 2x_{il}y_{lk} + (1 - 2v_{ik}) \right) x_{ij} \right]_{jk}, \\ Y(t) &= g(U_Y(t)). \end{aligned} \tag{14}$$

Eqn. (11) shows the partial derivative of x_{ij} exclusively depends on x_{il}, where $l \neq j$. It implies that the neuronal states in the same column of X can be updated synchronously. Similarly, Eqn. (12) shows the partial derivative of y_{jk} exclusively depends on y_{lk}, where $l \neq j$, implying that the states in the same row of Y can be updated synchronously. As a result, X and Y may be updated synchronously in r batches in DHN by updating states in the same column of X and the same row of Y.

Fig. 1 delineates the scheme of the proposed two-phase CNO-BMF algorithm. CNO-BMF starts with the first phase by running a population of DHNm's (3) synchronously for coarse searches and follows with the second phase by running DHNs (1) synchronously in batches for fine searches. The particle swarm optimization rule in (4) is used to initialize the neuronal states repetitively upon their local convergence.

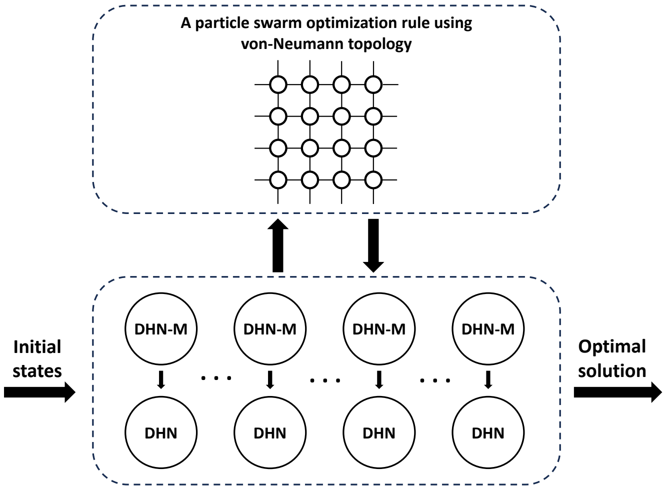


Figure 1: A schematic diagram of the CNO-BMF algorithm.

Algorithm 1 details the CNO-based binary matrix factorization. In the algorithm, Steps 6-8 are to asynchronously update X and Y according to the DHNm rule until the decline rate of the objective function value is lower than ϵ. Step 9 is to shuffle the ordered sets $\mathcal{B}_X$ and $\mathcal{B}_Y$ to introduce randomness for enhancing the diversity of solutions. Steps 10-13 are to update every column of X in a randomly ordered index set $\mathcal{B}_X$ and every row of Y in a randomly ordered index set $\mathcal{B}_Y$ alternately according to the DHN rule until convergence. Steps 14-16 and 18-23 are to update individual-best and population-best solutions, respectively. Steps 24-26 are to update X and Y according to the particle swarm optimization rule to escape from local minima in the global search of optima. In Step 27, the diversity of the N sets of solutions is measured according to (5). In Steps 28-30, the bit-flip mutation operator in (6) is performed if the diversity measure is below the preset threshold $\delta_{\min}$.

5. Experimental Results

5.1. Experiment Setups

In the experiments, the CNO-BMF parameters are set as follows. The population size N is set to 10, and termination criteria M is set to 50. The termination criteria for DHNm ϵ is set to 0.01. The diversity threshold $\delta_{\min}$ is set to 0.004, and the mutation probability ρ in (6) is set to 0.01. In the particle swarm optimization rule in (4), $c_0 = 1$, $c_1 = c_2 = 2$.

The experiments are based on eight benchmark datasets: Zoo[1], Lymp[2], Hepatitis[3], Wine[4], Audio[5], Votes[6], and Tic-

[1] `http://archive.ics.uci.edu/dataset/111/zoo`
[2] `https://archive.ics.uci.edu/dataset/63/lymphography`
[3] `https://archive.ics.uci.edu/dataset/46/hepatitis`
[4] `https://archive.ics.uci.edu/dataset/109/wine`
[5] `http://archive.ics.uci.edu/dataset/8/audiology+standardized`
[6] `https://archive.ics.uci.edu/dataset/105/congressional+`

Algorithm 1: CNO-BMF

Input: Data matrix V, population size N, termination criterion M, ordered batch index sets $\mathcal{B}_X = \{1, 2, \ldots, r\}$ and $\mathcal{B}_Y = \{1, 2, \ldots, r\}$, particle swarm optimization based parameters c_0, c_1, and c_2.

Output: X^* and Y^*.

1 For $k = 1, 2, \ldots, N$, generate random initial neuronal state matrices $X_k(0) \in \{0,1\}^{n\times r}$ and $Y_k(0) \in \{0,1\}^{r\times m}$, velocity matrices $V_k^X \in [-1,1]^{n\times r}, V_k^Y \in [-1,1]^{r\times m}$, set initial group-best matrix and initial individual-best matrices $X^* = \bar{X}_k = 0$ and $Y^* = \bar{Y}_k = 0$. Set $q = 0$;
2 **while** $q \leq M$ **do**
3   **for** $k = 1$ **to** N **do**
4         $u_k^X(0) \leftarrow X_k(0) \times m \times r$;
5         $u_k^Y(0) \leftarrow Y_k(0) \times n \times r$;
6   **while** $(f(X_k(t), Y_k(t)) - f(X_k(t+1), Y_k(t+1)))/f(X_k(t), Y_k(t)) < \epsilon$ **do**
7   Update $X_k(t)$ and $Y_k(t)$ according to (3) with $u_k^X(t+1)$ and $u_k^Y(t+1)$;
8   **end**
9   Shuffle the order of $\mathcal{B}_X$ and $\mathcal{B}_Y$;
10   **while** $X_k(t) \neq X_k(t+1)$ and $Y_k(t) \neq Y_k(t+1)$ **do**
11   Update every column of $X_k(t)$ in the order of $\mathcal{B}_X$ according to (13) ;
12   Update every row of $Y_k(t)$ in the order of $\mathcal{B}_Y$ according to (14) ;
13   **end**
14   **if** $f(X_k, Y_k) < f(\bar{X}_k, \bar{Y}_k)$ **then**
15             $\bar{X}_k \leftarrow X_k$ and $\bar{Y}_k \leftarrow Y_k$;
16   **end**
17   **end**
18     $(\hat{X}, \hat{Y}) = \arg\min\{f(X_1(t), Y_1(t)), \ldots, f(X_N(t), Y_N(t))\}$;
19   **if** $f(\hat{X}, \hat{Y}) < f(X^*, Y^*)$ **then**
20         $X^* \leftarrow \hat{X}$, $Y^* \leftarrow \hat{Y}$, and $q \leftarrow 0$;
21   **else**
22         $q \leftarrow q + 1$;
23   **end**
24   **for** $k = 1$ **to** N **do**
25   Update X_k and Y_k according to (4);
26   **end**
27   Compute $\delta(q)$ according to (5);
28   **if** $\delta(q) < \delta_{\min}$ **then**
29   Perform the bit-flip mutation according to (6);
30   **end**
31 **end**

tac-Toe[7], ORL (Samaria & Harter (1994)), with their major parameters listed in Table 1.

The proposed CNO-BMF algorithm is compared with seven prevailing algorithms for BMF: thresholding method of BMF (BMF-TH) (Zhang et al. (2010)), the penalty objective formulation (referred to as ZH) (Zhang et al. (2007)), the greedy algorithm for k-BMF (k-Greedy) (Kovacs et al. (2021)), binary matrix factorization via column generation (BMF-CG-MIP(1)) (Kovacs et al. (2021)), binary matrix factorization via column generation with Frobenius norm (BMF-CG-MIP$_F$) (Kovacs et al. (2021)), and genetic algorithm for binary matrix factorization (BMF-GA) (Snášel et al. (2008)). The code of BMF-TH is obtained from the Github[8] of the first author of Zhang et al. (2010). The code of ZH is obtained from a Python package: PyMF[9]. The codes of k-Greedy, BMF-CG-MIP(1), and BMF-CG-MIP$_F$ are obtained from the Github[10].

5.2. Neurodynamic Behaviors

Fig. 2 illustrates eight snapshots of the convergent behaviors of the objective function $f(X, Y)$ in (7) resulting from DHNm's and DHNs in the inner-loop of CNO-BMF on the eight datasets, where the blue dotted lines are for the phase of DHNm updating (Steps 6-8) and the red line is for the phase of DHN updating (Steps 10-13). Fig. 2 shows that the values of objective function monotonically decrease and reach stationary points within 210 iterations. Fig. 3 depicts the convergent behaviors of $f(X, Y)$ using CNO-BMF on the eight datasets, where the red envelopes depict the objective functions of group-best solutions X^* and Y^*. It shows that the objective function values monotonically decline, and CNO-BMF converges within 350 iterations.

5.3. Ablation Studies

In the ablation studies, the performance of the two-phase CNO-BMF algorithm with DHNm and DHN is compared with those of two one-phase ones (with DHNm or DHN only). Fig. 4 illustrates eight snapshots of the convergent behaviors of the objective function $f(X, Y)$ in (7) resulting from DHNm on the eight datasets. As shown in Fig. 4, DHNm takes much more iterations to converge (i.e., ranging from 5700 to 980000 iterations) than DHN in the inner-loop of CNO-BMF (i.e., about 260 iterations) as shown in Fig. 2.

Fig 5 depicts the Monte Carlo test results using CNO-BMF with DHN (denoted as CNO-BMF/DHN) and CNO-BMF with DHNm-DHN (denoted as CNO-BMF/DHNm-DHN) with three values of rank r on the eight datasets.

---

voting+records

[7] https://archive.ics.uci.edu/dataset/101/tic+tac+toe+endgame

[8] https://github.com/ZhongYuanZhang/BMF

[9] https://github.com/rikkhill/pymf

[10] https://github.com/kovacsrekaagnes/rank_k_Binary_Matrix_Factorisation

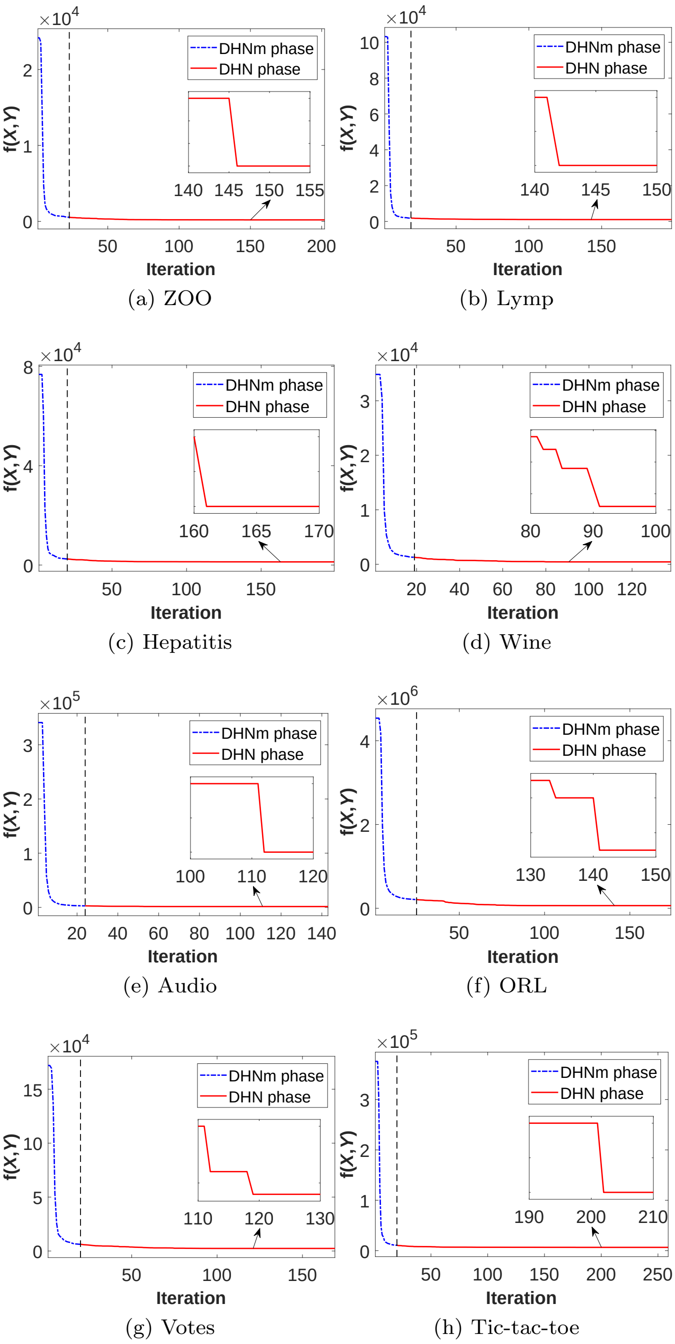


Figure 2: Snapshots of the objective function values of $f(X, Y)$ in (7) in the inner-loop of CNO-BMF on the eight datasets, where the blue dotted line is in the phase of DHNm updating (Steps 6-8), and the red line is the phase of DHN updating (Steps 10-13).

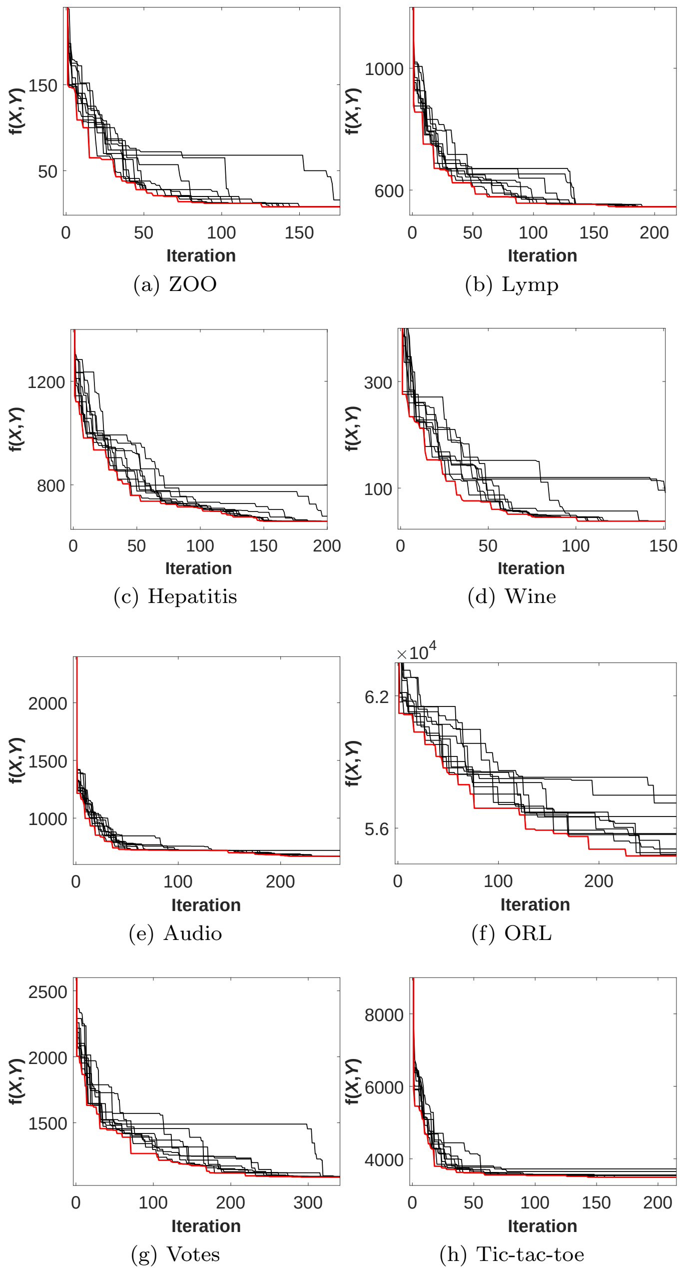
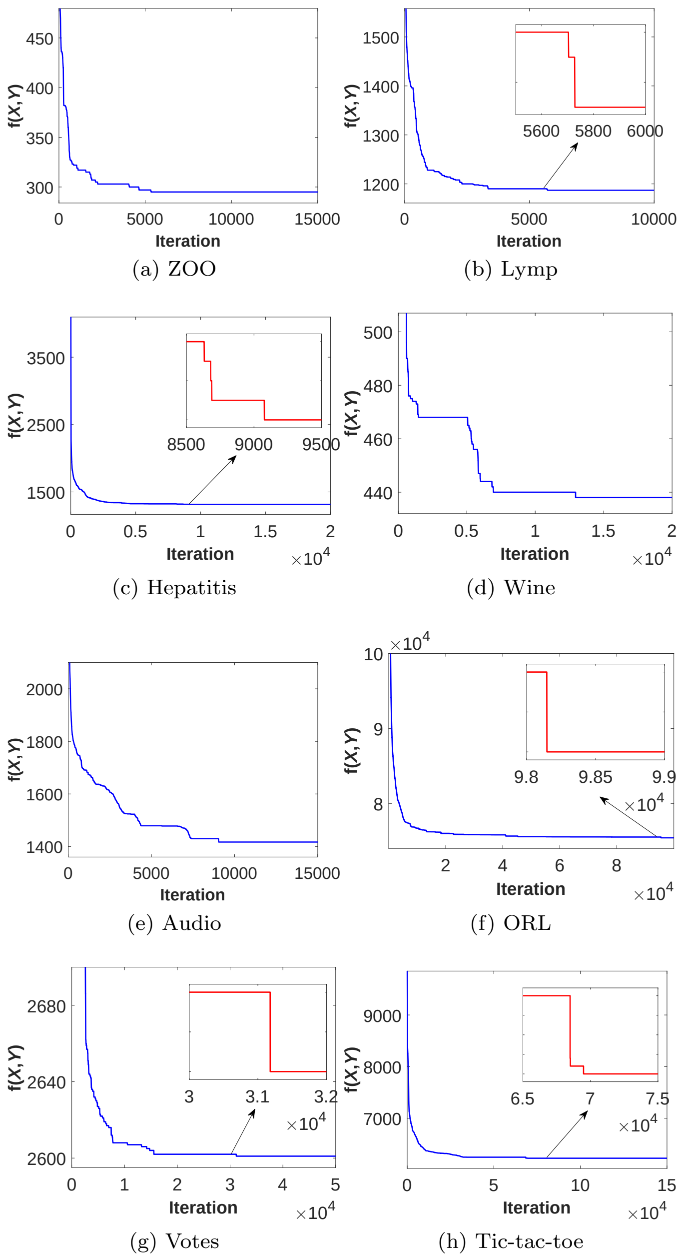


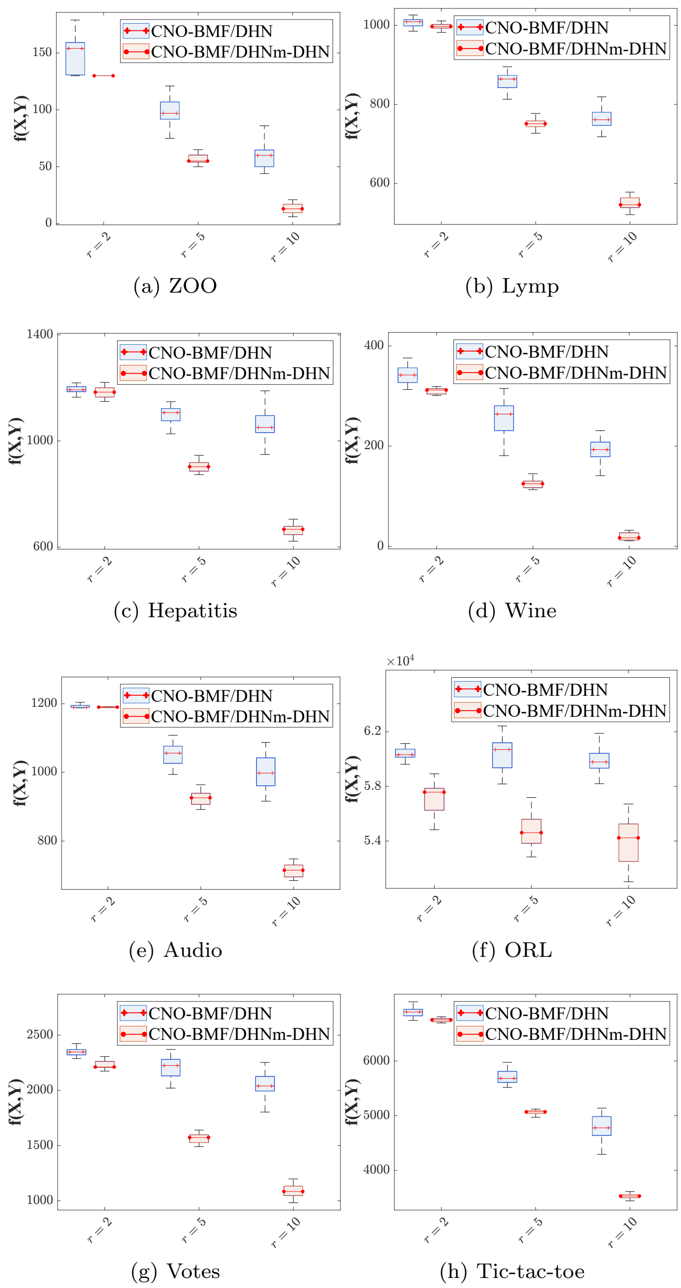
Figure 3: The convergent behavior of CNO-BMF.



(a) ZOO (b) Lymp (c) Hepatitis (d) Wine (e) Audio (f) ORL (g) Votes (h) Tic-tac-toe

Figure 4: Snapshots of the objective function values of $f(X,Y)$ in (7) resulting from DHNm on the eight datasets.

As shown in Fig 5, CNO-BMF/DHNm-DNN consistently outperforms CNO-BMF/DHN in terms of objective function value, especially for large values of r.



(a) ZOO (b) Lymp (c) Hepatitis (d) Wine (e) Audio (f) ORL (g) Votes (h) Tic-tac-toe

Figure 5: Monte Carlo test results using CNO-BMF/DHN and CNO-BMF/DHNm-DNN with three values of r on the eight datasets.

5.4. Performance Comparisons

In CNO-BMF, there are two hyper-parameters: the DHN population size N and the minimum number of consecutive iterations M without further improvement as the termination criterion. Fig 6 depicts the Monte Carlo test results using CNO-BMF with several values of N and M on ZOO and Lymp. As shown in Fig 6, with increasing values of N or M, the value of $f(X, Y)$ resulting from CNO-BMF declines. The objective function values always reach their minima in all 100 runs if $M \geq 30$ and $N \geq 10$ using CNO-BMF on ZOO ($r = 2$) and Lymp ($r = 2$). It shows that CNO-BMF can almost ensure convergence to global optima, provided that the values of N and M are large enough depending on the complexity of the problem.

Table 1 records the mean values and standard deviations of the objective function values using CNO-BMF ($N = 10$ and $M = 50$) and the six baselines over 50 runs with random initialization on the eight datasets with numerous rank values ($r = 2, 3, 5, 10, 15$). Table 1 shows that CNO-BMF obtains the best results among the seven methods in terms of the mean values of errors on the eight datasets with various rank values r. In addition, it also shows that the larger the rank value, the smaller the factorization error in the results obtained using CNO-BMF.

6. Pattern Discovery

Pattern discovery is to identify meaningful patterns or structures in a given matrix. It is an important task in various fields, including data mining and machine learning (Koyuturk et al. (2005); Jiang & Heath (2013)). BMF is an approach for discovering binary patterns. It involves finding two binary matrices of a low rank (i.e., dominant features) to minimize the difference between their matrix product (i.e., $V_r = XY$) and a given binary matrix (i.e., V) (Koyuturk et al. (2005); Jiang & Heath (2013); Shen et al. (2009a)). By approximating a given matrix, BMF aims to capture the most dominant features that may represent patterns, whereas noise may be disregarded in the product of the factorized matrices XY (Shen et al. (2009b); Lucchese et al. (2010); Lu et al. (2020); Liang et al. (2020)).

Consider 200×80 binary matrix with implanted patterns (i.e., V) called PD1 shown in Fig. 7a as presented in Lu et al. (2020), where a black point indicates an element with the value of 1. As in Lu et al. (2020), each element in V is flipped with probability 0.05, resulting in a noised matrix $\tilde{V}$ shown in Fig. 7b, where $r = 5$. Figs. 7c-7i show matrices resulting from factorized matrices (i.e., $Z = XY$) using CNO-BMF and the six baselines with $r = 5$ on PD1. As shown in Fig. 7, CNO-BMF is able to capture the underlying seven patterns in the given matrix better than the six baselines on PD1.

To quantify the performance of CNO-BMF and six baselines on various datasets with various rank values, two additional datasets PD2 in Koyuturk et al. (2005) and PD3 in Jiang & Heath (2013) are used in the experiments, where noise points are added in the implanted pattern matrix according to the literature. Table 2 records the mean values and standard deviations of the pattern discovery error (i.e., $||\tilde{V} - XY||_F$), precision, and recall using CNO-BMF ($N = 10$ and $M = 50$) and six baselines on the three datasets (i.e., PD1, PD2, and PD3) with various rank values (i.e., $r = 2, 4, 6$), where $\tilde{V}$ represents the implanted pattern matrix. As shown in Table 2, the mean values of the pattern discovery error decrease with the increasing rank values using CNO-BMF and the six baselines. CNO-BMF consistently outperforms the baselines, in terms of the mean values of pattern discovery error and most of the mean values of precision and recall on the three datasets and various rank values. It indicates the ability of CNO-BMF to capture meaningful patterns in binary matrices accurately.

7. Concluding Remarks

This paper presents a binary matrix factorization algorithm based on collaborative neurodynamic optimization. The proposed algorithm statistically outperforms the baselines owing to the combined use of a more powerful discrete Hopfield network and a more effective collaborative neurodynamic optimization framework. Further investigations may aim at developing a more efficient binary matrix factorization algorithm assisted by deep learning and reinforcement learning, and customizing the binary matrix factorization algorithm in specific application domains such as associate rule mining, market basket data clustering, and document clustering.

References

Beckerleg, M., & Thompson, A. (2020). A divide-and-conquer algorithm for binary matrix completion. *Linear Algebra and its Applications*, *601*, 113–133.

Cernuschi-Frías, B. (1989). Partial simultaneous updating in Hopfield memories. *IEEE Transactions on Systems, Man, and Cybernetics*, *19*, 887–888.

Chandran, S., Issac, D., & Karrenbauer, A. (2017). On the parameterized complexity of biclique cover and partition. In *11th International Symposium on Parameterized and Exact Computation* (pp. 11:1–11:13). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik volume 63.

Che, H., & Wang, J. (2018). A nonnegative matrix factorization algorithm based on a discrete-time projection neural network. *Neural Networks*, *103*, 63–71.

Che, H., & Wang, J. (2019). A collaborative neurodynamic approach to global and combinatorial optimization. *Neural Networks*, *114*, 15 – 27.

Che, H., & Wang, J. (2021). A two-timescale duplex neurodynamic approach to mixed-integer optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *32*, 36–48.

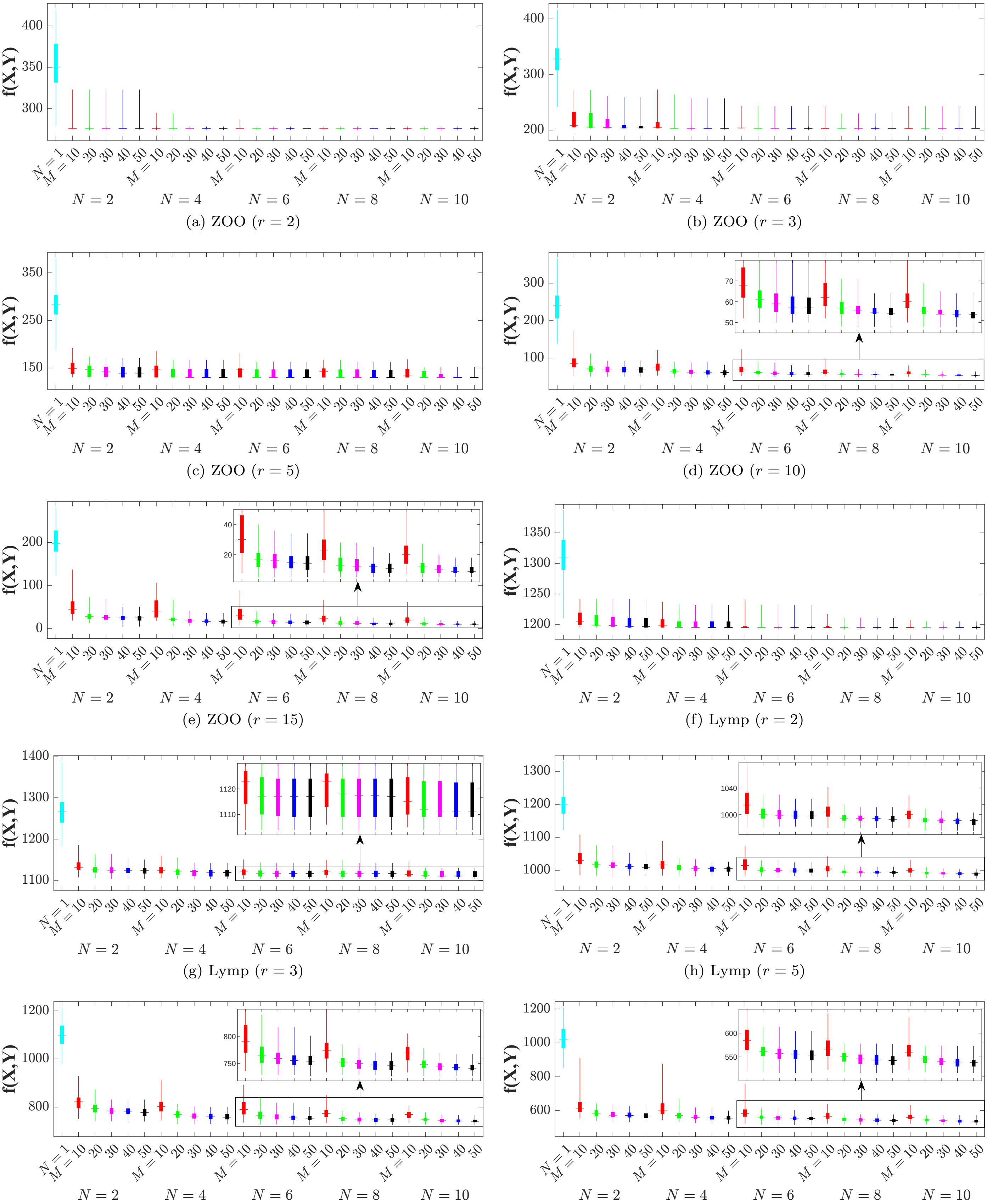
Che, H., Wang, J., & Cichocki, A. (2022). Sparse signal reconstruction via collaborative neurodynamic optimization. *Neural Networks*, *154*, 255–269.

Che, H., Wang, J., & Cichocki, A. (2023). Bicriteria sparse nonnegative matrix factorization via two-timescale duplex neurodynamic optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *34*, 4881–4891.

Dan, C., Hansen, K. A., Jiang, H., Wang, L., & Zhou, Y. (2018). Low Rank Approximation of Binary Matrices: Column Subset Selection and Generalizations. In *43rd International Symposium on Mathematical Foundations of Computer Science* (pp. 41:1–41:16). Schloss Dagstuhl – Leibniz-Zentrum für Informatik volume 117.

Diop, M., Larue, A., Miron, S., & Brie, D. (2017). A post-nonlinear mixture model approach to binary matrix factorization. In *25th European Signal Processing Conference* (pp. 321–325). IEEE.

Fu, Y., Jiang, N., & Sun, H. (2010). Binary matrix factorization and consensus algorithms. In *2010 International Conference on Electrical and Control Engineering* (pp. 4563–4567). IEEE.



(a) ZOO ($r = 2$) (b) ZOO ($r = 3$)

(c) ZOO ($r = 5$) (d) ZOO ($r = 10$)

(e) ZOO ($r = 15$) (f) Lymp ($r = 2$)

(g) Lymp ($r = 3$) (h) Lymp ($r = 5$)

(i) Lymp ($r = 10$) (j) Lymp ($r = 15$)

Figure 6: Monte Carlo test results using CNO-BMF with several values of N and M on ZOO and Lymp.

Table 1: The mean values and standard deviations of the objective function values using CNO-BMF ($N = 10$ and $M = 50$) and the six baselines on Zoo, Lymp, Hepatitis, Wine, and Audio with numerous rank values ($r = 2, 3, 5, 10, 15$).

Dataset ($n \times m$)	rank r	ZH	BMF-TH	k-Greedy	BMF-CG-MIP(1)	BMF-CG-MIP$_F$	BMF-GA	CNO-BMF (herein)
Zoo (101 × 17)	2	308.9600 ± 32.9400	279.0000 ± 0.0000	404.5200 ± 35.3600	295.4000 ± 2.3800	297.0000 ± 0.0000	406.9200 ± 9.1500	**276.0000 ± 0.0000**
	3	340.4400 ± 59.8200	225.7200 ± 2.2800	406.3200 ± 56.9500	228.6400 ± 8.7700	232.4800 ± 8.5100	432.5200 ± 15.3800	**203.0000 ± 0.0000**
	5	391.3200 ± 52.0000	148.2800 ± 10.2500	406.2800 ± 85.4800	147.0800 ± 4.3800	152.0000 ± 8.1300	490.5600 ± 14.6200	**135.8800 ± 8.3500**
	10	543.3600 ± 35.4100	124.5200 ± 23.1800	407.0800 ± 98.5900	282.5600 ± 45.3300	316.3200 ± 44.2900	763.2400 ± 32.5700	**57.0800 ± 6.0900**
	15	620.1600 ± 39.4700	47.5600 ± 26.5600	401.4000 ± 97.6700	730.0400 ± 179.5100	637.7200 ± 146.1700	1830.0400 ± 157.2600	**12.0800 ± 4.7100**
Lymp (148 ×44)	2	1346.7200 ± 68.3000	1271.3600 ± 0.8100	1389.3600 ± 56.8800	1277.8400 ± 15.5700	1341.7600 ± 57.9800	1503.3200 ± 15.3700	**1195.4000 ± 1.9800**
	3	1305.4400 ± 71.7200	1229.3200 ± 3.3100	1449.1600 ± 77.7700	1283.0800 ± 26.4400	1423.6000 ± 65.8800	1643.7200 ± 27.7100	**1112.9600 ± 6.9600**
	5	1407.5600 ± 78.4700	1205.9200 ± 30.9900	1580.5200 ± 162.7200	1330.9600 ± 82.0800	1654.4400 ± 136.3500	2385.6400 ± 66.1500	**995.8800 ± 8.6900**
	10	1650.6400 ± 89.7300	1179.2000 ± 45.4300	1721.9600 ± 187.4700	1430.6000 ± 159.4100	2033.7200 ± 297.0200	8471.9200 ± 357.7300	**749.3600 ± 13.0400**
	15	1811.5200 ± 60.0000	1024.8000 ± 69.1000	1796.0800 ± 207.9300	1424.7600 ± 177.8200	1954.8000 ± 372.3200	23064.4000 ± 1056.4200	**545.6000 ± 13.4800**
Hepatitis (155 ×38)	2	1632.8800 ± 67.1700	1446.0000 ± 0.0000	1699.9600 ± 67.7700	1466.0400 ± 14.1300	1512.8400 ± 20.0200	1939.0400 ± 28.2600	**1385.6000 ± 1.0800**
	3	1763.6800 ± 92.8800	1492.1200 ± 29.4000	1824.7200 ± 123.7500	1850.6400 ± 91.3500	2219.0000 ± 370.2700	2080.0800 ± 24.7300	**1295.2000 ± 19.1300**
	5	2193.4000 ± 147.2800	1549.1200 ± 39.1600	2032.6000 ± 151.2500	2086.3600 ± 220.1700	3255.2400 ± 583.5400	2484.1600 ± 45.1300	**1183.7200 ± 21.3000**
	10	2609.1600 ± 83.4200	1669.9200 ± 82.6600	2341.3200 ± 167.8700	2540.9600 ± 279.7800	4892.7200 ± 773.6400	5923.5200 ± 327.9000	**907.1200 ± 25.6900**
	15	2746.3200 ± 38.2500	1601.8800 ± 131.4600	2523.0800 ± 182.5900	2936.3200 ± 598.0300	5195.5600 ± 1475.7500	16426.3200 ± 898.4100	**662.4800 ± 21.7100**
Wine (178 ×16)	2	755.6400 ± 33.0600	629.8800 ± 0.6000	704.4400 ± 84.7600	636.7600 ± 11.0200	648.0800 ± 13.5900	1161.5000 ± 203.0700	**616.0000 ± 0.0000**
	3	615.9600 ± 58.6600	426.8400 ± 5.5500	737.4800 ± 137.9900	490.9600 ± 6.9700	504.7200 ± 16.6300	1185.7000 ± 178.9000	**411.2400 ± 0.6600**
	5	792.4400 ± 106.6400	421.2000 ± 67.3200	788.6000 ± 143.6800	618.7200 ± 96.3700	925.2800 ± 195.0400	1225.3000 ± 138.6400	**311.2000 ± 7.7900**
	10	1207.1600 ± 95.2700	280.5600 ± 62.5800	857.4800 ± 154.9600	464.5200 ± 97.0400	646.2800 ± 120.2800	1368.5600 ± 31.3800	**125.5200 ± 9.7600**
	15	1387.9200 ± 75.4200	117.6000 ± 64.2100	852.4400 ± 156.0900	682.4000 ± 59.4000	628.7600 ± 86.7000	2012.3000 ± 676.6300	**20.3600 ± 7.6900**
Audio (226 ×94)	2	1563.0400 ± 77.8700	1507.0000 ± 0.0000	1570.5200 ± 60.4100	1510.2800 ± 0.9800	1510.2800 ± 0.9800	3093.6400 ± 115.5800	**1503.0000 ± 0.0000**
	3	1537.5200 ± 49.5500	1368.8800 ± 2.3300	1553.0000 ± 68.6800	1469.4000 ± 36.1400	1493.1600 ± 38.5600	5284.4000 ± 154.1300	**1337.0000 ± 0.0000**
	5	1606.6000 ± 117.7300	1316.8000 ± 14.9100	1544.2000 ± 106.2400	1653.8000 ± 124.1200	1838.3200 ± 179.0500	12489.0400 ± 483.2800	**1190.9200 ± 5.4500**
	10	1971.1600 ± 112.4600	1274.5600 ± 85.0000	1536.8400 ± 118.2100	1457.8800 ± 114.5100	1945.4800 ± 253.1100	53279.8000 ± 1608.3800	**925.4400 ± 24.0400**
	15	2131.6000 ± 85.7200	1050.5600 ± 76.2600	1555.5600 ± 128.2500	1283.6800 ± 141.6000	1511.3200 ± 229.5600	131283.6000 ± 3888.4800	**714.1600 ± 18.7900**
Votes (434 ×32)	2	3053.2800 ± 1.5400	2963.0000 ± 0.0000	2938.2800 ± 2.5900	**2926.0000 ± 0.0000**	**2926.0000 ± 0.0000**	5393.8400 ± 104.8700	**2926.0000 ± 0.0000**
	3	3107.0000 ± 241.4200	2846.5200 ± 46.0800	3095.8800 ± 140.9000	3091.0400 ± 136.1300	4128.8400 ± 537.6300	5711.6000 ± 52.0100	**2645.2000 ± 26.0200**
	5	3613.2800 ± 456.6000	2755.6400 ± 64.6300	3355.7200 ± 166.7800	3362.0000 ± 198.2300	5795.2400 ± 1330.5200	6319.9200 ± 48.0500	**2231.3200 ± 35.6100**
	10	5597.3600 ± 414.5400	2640.6400 ± 332.4800	3898.2400 ± 248.3600	4104.4800 ± 673.2300	7121.3600 ± 1194.8000	13342.1200 ± 652.9000	**1564.7600 ± 43.1300**
	15	6338.4400 ± 332.0100	2939.2800 ± 400.8900	4331.4400 ± 298.9400	4775.4400 ± 826.4200	8443.1200 ± 2477.3200	37495.6000 ± 1383.1700	**1088.4400 ± 62.3200**
Tic-tac-toe (958 ×30)	2	8881.2800 ± 199.9800	8440.0000 ± 0.0000	8442.0000 ± 120.4200	8252.9200 ± 49.6400	8263.6400 ± 52.1600	9249.8400 ± 15.1100	**8167.5200 ± 12.0200**
	3	8504.2800 ± 161.8600	8116.8000 ± 103.7700	8097.6000 ± 111.9800	7750.6400 ± 59.6200	7926.4000 ± 85.2600	9336.1600 ± 14.8300	**7636.6000 ± 25.3600**
	5	7922.8800 ± 220.2800	7885.2800 ± 191.2100	7720.2000 ± 143.8800	7316.3200 ± 148.9700	8432.4800 ± 503.9100	10352.6400 ± 212.5600	**6750.0000 ± 34.1200**
	10	8566.1600 ± 403.4000	6834.2800 ± 297.6300	7555.5200 ± 191.5100	5530.1200 ± 133.6300	6543.0800 ± 1355.0200	28055.0800 ± 1306.9000	**5064.5200 ± 35.2900**
	15	9580.0000 ± 0.0000	6146.0400 ± 388.4400	7689.8400 ± 243.0100	3704.0000 ± 114.6200	3810.3600 ± 193.4600	82211.0400 ± 3971.7000	**3535.4400 ± 47.8400**
ORL (400 ×1024)	2	343534.0000 ± 0.0000	89138.4400 ± 369.4300	161677.3600 ± 1407.4200	64447.0000 ± 0.0000	161677.3600 ± 1407.4200	649147.2400 ± 443002.1300	**60744.6000 ± 39.8100**
	3	343534.0000 ± 0.0000	120363.7200 ± 508.9600	175107.8400 ± 12807.1300	67794.5200 ± 2599.4200	175107.8400 ± 12807.1300	659843.8600 ± 432201.0100	**57814.7200 ± 717.4000**
	5	343534.0000 ± 0.0000	152493.0400 ± 13534.7400	187925.9200 ± 9510.5000	70041.1200 ± 2233.9600	812804.1200 ± 206657.2400	701213.3000 ± 390437.6600	**57195.4400 ± 1268.5400**
	10	343534.0000 ± 0.0000	163715.5600 ± 9916.0300	189480.1600 ± 7664.4700	68726.9600 ± 2912.7900	916988.3200 ± 224914.0800	1087514.4000 ± 18027.6700	**54739.6800 ± 1141.1100**
	15	343534.0000 ± 0.0000	183350.6400 ± 17425.8900	189583.6000 ± 7705.1700	68213.1200 ± 2579.2600	1289046.2400 ± 696388.0300	1959888.9600 ± 881819.5600	**54003.2000 ± 1652.7000**

Galán-Marín, G., & Muñoz-Pérez, J. (2001). Design and analysis of maximum Hopfield networks. *IEEE Transactions on Neural Networks*, *12*, 329–339.

Gillis, N., & Vavasis, S. A. (2018). On the complexity of robust pca and ℓ_1-norm low-rank matrix approximation. *Mathematics of Operations Research*, *43*, 1072–1084.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554 – 2558.

Hopfield, J. J., & Tank, D. W. (1985). Neural computation of decisions in optimization problems. *Biological Cybernetics*, *52*, 141–152.

Hopfield, J. J., & Tank, D. W. (1986). Computing with neural circuits - a model. *Science*, *233*, 625–633.

Huang, B., Liu, Y., Jiang, Y.-L., & Wang, J. (2024). Two-timescale projection neural networks in collaborative neurodynamic approaches to global optimization and distributed optimization. *Neural Networks*, *169*, 83–91.

Jia, W., Huang, T., & Qin, S. (2024). A collective neurodynamic penalty approach to nonconvex distributed constrained optimization. *Neural Networks*, *171*, 145–158.

Jiang, P., & Heath, M. T. (2013). Pattern discovery in high dimensional binary data. In *IEEE 13th International Conference on Data Mining Workshops* (pp. 474–481). IEEE.

Jiang, P., Peng, J., Heath, M., & Yang, R. (2014). A clustering approach to constrained binary matrix factorization. *Data Mining and Knowledge Discovery for Big Data: Methodologies, Challenge and Opportunities*, (pp. 281–303).

Jin, L., Su, Z., Fu, D., & Xiao, X. (2024). Coevolutionary neural solution for nonconvex optimization with noise tolerance. *IEEE Transactions on Neural Networks and Learning Systems*, . In press.

Ju, X., Li, C., He, X., & Feng, G. (2024a). A proximal dynamic approach to equilibrium problems with finite-time convergence. *IEEE Transactions on Automatic Control*, . In press.

Ju, X., Yang, X., Feng, G., & Che, H. (2023). Neurodynamic optimization approaches with finite/fixed-time convergence for absolute value equations. *Neural Networks*, *165*, 971–981.

Ju, X., Yuan, S., Yang, X., & Shi, P. (2024b). Fixed-time neurodynamic optimization algorithms and application to circuits design. *IEEE Transactions on Circuits and Systems I: Regular Papers*, . In press.

Kennedy, J., & Mendes, R. (2002). Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation.* (pp. 1671–1676). volume 2.

Kovacs, R. A., Gunluk, O., & Hauser, R. A. (2021). Binary matrix factorisation via column generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 3823–3831). volume 35.

Koyutürk, M., & Grama, A. (2003). Proximus: a framework for analyzing very high dimensional discrete-attributed datasets. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 147–156).

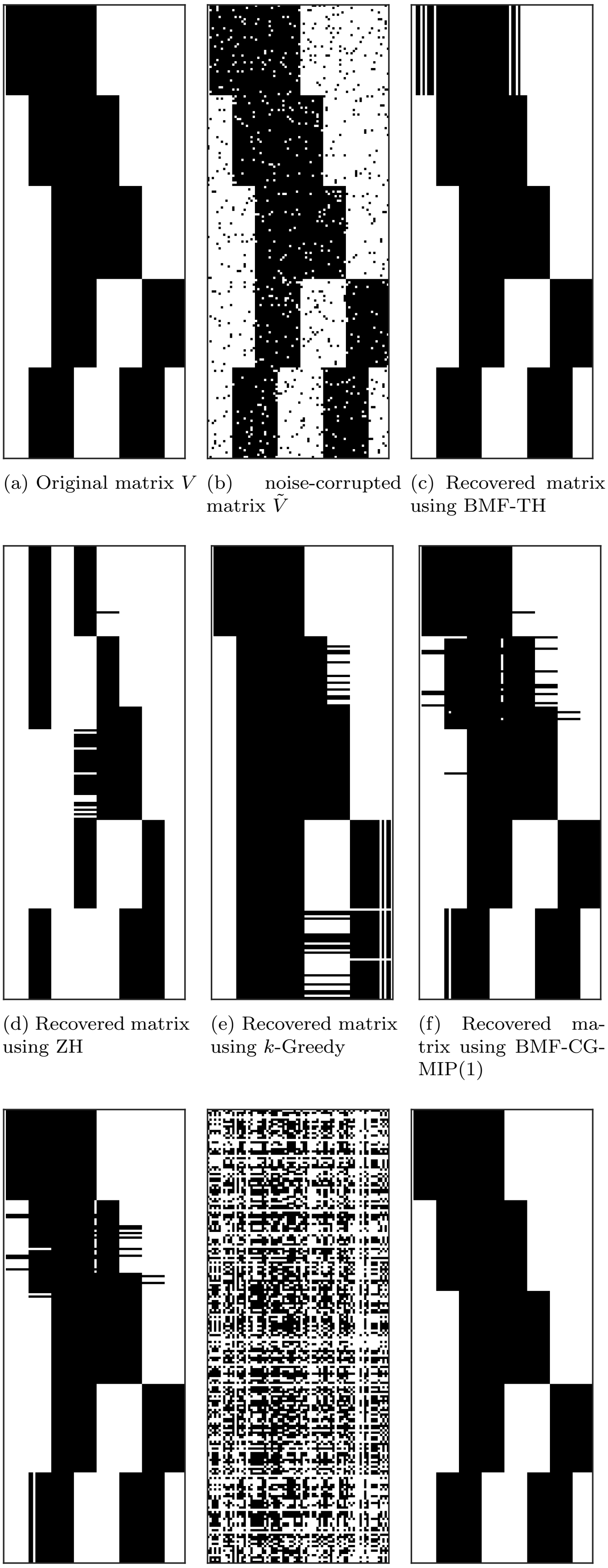
Koyutürk, M., Grama, A., & Ramakrishnan, N. (2002). Algebraic techniques for analysis of large discrete-valued datasets. In *Principles of Data Mining and Knowledge Discovery: 6th European Conference* (pp. 311–324). Springer.

Koyuturk, M., Grama, A., & Ramakrishnan, N. (2005). Compression, clustering, and pattern discovery in very high-dimensional discrete-attribute data sets. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 447–461.

Koyutürk, M., Grama, A., & Ramakrishnan, N. (2006). Nonorthogonal decomposition of binary matrices for bounded-error data compression and analysis. *ACM Transactions on Mathematical Software*, *32*, 33–69.

Kumar, R., Panigrahy, R., Rahimi, A., & Woodruff, D. (2019). Faster algorithms for binary matrix factorization. In *International Conference on Machine Learning* (pp. 3551–3559). PMLR.

Lee, D.-L. (1999). New stability conditions for Hopfield networks in



(a) Original matrix V (b) noise-corrupted matrix $\tilde{V}$ (c) Recovered matrix using BMF-TH

(d) Recovered matrix using ZH (e) Recovered matrix using k-Greedy (f) Recovered matrix using BMF-CG-MIP(1)

(g) Recovered matrix using BMF-CG-MIP$_F$ (h) Recovered matrix using BMF-GA (i) Recovered matrix using CNO-BMF

Figure 7: Original matrix, noise-corrupted matrix, and recovered matrices from factorized matrices (i.e., XY) using CNO-BMF and the six baselines ($r = 5$) on PD1.

Table 2: The mean values and standard deviations of the pattern discovery error, precision, and recall using CNO-BMF and six baselines on the three datasets with various rank values.

Datasets	Rank	Method	$\|\|V - XY\|\|_F$	Precision	Recall
PD1	$r = 2$	ZH	10018.5200 ± 1531.2859	0.3448 ± 0.0874	0.2399 ± 0.0481
		BMF-TH	3151.9600 ± 0.2000	0.8099 ± 0.0006	**0.7894 ± 0.0010**
		k-Greedy	13933.2400 ± 896.2378	0.1516 ± 0.0283	0.1461 ± 0.0390
		BMF-CG-MIP(1)	9409.0000 ± 0.0000	0.3925 ± 0.0000	0.3273 ± 0.0000
		BMF-CG-MIP$_F$	9409.0000 ± 0.0000	0.3925 ± 0.0000	0.3273 ± 0.0000
		BMF-GA	5889.9200 ± 118.4936	0.6944 ± 0.0296	0.4384 ± 0.0252
		CNO-BMF (herein)	**3143.6800 ± 6.5302**	**0.8297 ± 0.0140**	0.7622 ± 0.0200
	$r = 4$	ZH	12704.1200 ± 1298.1619	0.1651 ± 0.0539	0.1132 ± 0.0397
		BMF-TH	814.5600 ± 17.8397	0.9427 ± 0.0084	**0.9558 ± 0.0093**
		k-Greedy	21295.9600 ± 1325.2635	0.1503 ± 0.0260	0.1531 ± 0.0284
		BMF-CG-MIP(1)	10123.2400 ± 98.5111	0.3518 ± 0.0051	0.3499 ± 0.0028
		BMF-CG-MIP$_F$	10069.0000 ± 0.0000	0.3546 ± 0.0000	0.3515 ± 0.0000
		BMF-GA	7578.2000 ± 124.4495	0.4803 ± 0.0129	0.4081 ± 0.0176
		CNO-BMF (herein)	**805.5200 ± 6.4622**	**0.9550 ± 0.0042**	0.9433 ± 0.0046
	$r = 6$	ZH	14821.4000 ± 1884.4967	0.1048 ± 0.0345	0.0628 ± 0.0206
		BMF-TH	884.8800 ± 363.2763	0.9098 ± 0.0613	0.8979 ± 0.0473
		k-Greedy	18191.1200 ± 2055.9034	0.1158 ± 0.0262	0.1126 ± 0.0239
		BMF-CG-MIP(1)	18240.0000 ± 0.0000	0.1688 ± 0.0000	0.2248 ± 0.0000
		BMF-CG-MIP$_F$	18240.0000 ± 0.0000	0.1688 ± 0.0000	0.2248 ± 0.0000
		BMF-GA	10772.0800 ± 286.0868	0.3344 ± 0.0099	0.4302 ± 0.0157
		CNO-BMF (herein)	**0.9600 ± 3.3226**	**0.9999 ± 0.0003**	**1.0000 ± 0.0001**
PD2	$r = 2$	ZH	5289.2400 ± 496.7100	0.3312 ± 0.0329	0.2808 ± 0.0254
		BMF-TH	1823.0000 ± 0.0000	0.9049 ± 0.0000	**0.6243 ± 0.0000**
		k-Greedy	4489.4000 ± 287.1283	0.4261 ± 0.0508	0.2810 ± 0.0515
		BMF-CG-MIP(1)	4797.0000 ± 0.0000	0.3566 ± 0.0000	0.1980 ± 0.0000
		BMF-CG-MIP$_F$	4797.0000 ± 0.0000	0.3566 ± 0.0000	0.1980 ± 0.0000
		BMF-GA	3145.6800 ± 117.8706	0.7576 ± 0.0183	0.3502 ± 0.0450
		CNO-BMF (herein)	**1814.0000 ± 0.0000**	**0.9337 ± 0.0000**	0.6037 ± 0.0000
	$r = 4$	ZH	7337.6400 ± 1174.8855	0.1485 ± 0.0775	0.1084 ± 0.0525
		BMF-TH	534.1600 ± 17.5919	0.9625 ± 0.0081	0.8956 ± 0.0071
		k-Greedy	7262.6800 ± 903.2486	0.1938 ± 0.0390	0.1968 ± 0.0403
		BMF-CG-MIP(1)	9507.0000 ± 0.0000	0.0756 ± 0.0000	0.0765 ± 0.0000
		BMF-CG-MIP$_F$	9643.0000 ± 0.0000	0.1036 ± 0.0000	0.1070 ± 0.0000
		BMF-GA	3943.2800 ± 73.3965	0.5188 ± 0.0159	0.3257 ± 0.0265
		CNO-BMF (herein)	**520.3600 ± 35.9199**	**0.9734 ± 0.0106**	**0.8964 ± 0.0187**
	$r = 6$	ZH	8320.9600 ± 1171.4189	0.1214 ± 0.0363	0.0812 ± 0.0139
		BMF-TH	701.1200 ± 176.7895	0.9405 ± 0.0336	0.8357 ± 0.0405
		k-Greedy	7870.4000 ± 1172.6092	0.1898 ± 0.0581	0.2015 ± 0.0565
		BMF-CG-MIP(1)	8238.4800 ± 419.8425	0.1688 ± 0.0020	0.1884 ± 0.0282
		BMF-CG-MIP$_F$	10059.0000 ± 0.0000	0.1330 ± 0.0000	0.1673 ± 0.0000
		BMF-GA	5230.7200 ± 127.1402	0.3710 ± 0.0093	0.4045 ± 0.0229
		CNO-BMF (herein)	**67.5200 ± 13.9556**	**0.9978 ± 0.0009**	**0.9839 ± 0.0034**
PD3	$r = 2$	ZH	12972.8400 ± 1811.7351	0.3164 ± 0.0812	0.2198 ± 0.0571
		BMF-TH	4149.0000 ± 0.0000	**1.0000 ± 0.0000**	0.6043 ± 0.0000
		k-Greedy	15796.4400 ± 775.1617	0.2117 ± 0.0431	0.2228 ± 0.0634
		BMF-CG-MIP(1)	16118.0000 ± 0.0000	0.2033 ± 0.0000	0.2199 ± 0.0000
		BMF-CG-MIP$_F$	16118.0000 ± 0.0000	0.2033 ± 0.0000	0.2199 ± 0.0000
		BMF-GA	7910.6400 ± 87.3508	0.6746 ± 0.0177	0.4522 ± 0.0169
		CNO-BMF (herein)	**3805.5200 ± 2.4000**	0.8024 ± 0.0217	**0.8471 ± 0.0282**
	$r = 4$	ZH	19038.2800 ± 2605.8536	0.1349 ± 0.0578	0.1240 ± 0.0508
		BMF-TH	1834.6800 ± 438.8687	0.9578 ± 0.0196	0.8505 ± 0.0350
		k-Greedy	20286.4800 ± 2062.2963	0.1128 ± 0.0431	0.1008 ± 0.0338
		BMF-CG-MIP(1)	20347.0000 ± 0.0000	0.0715 ± 0.0000	0.0788 ± 0.0000
		BMF-CG-MIP$_F$	20347.0000 ± 0.0000	0.0715 ± 0.0000	0.0788 ± 0.0000
		BMF-GA	10228.4400 ± 174.9076	0.4738 ± 0.0134	0.4054 ± 0.0218
		CNO-BMF (herein)	**747.2000 ± 16.0000**	**0.9687 ± 0.0089**	**0.9522 ± 0.0085**
	$r = 6$	ZH	23212.7600 ± 3481.1474	0.0656 ± 0.0238	0.0488 ± 0.0183
		BMF-TH	558.0000 ± 284.5306	0.9717 ± 0.0338	0.9478 ± 0.0282
		k-Greedy	19163.7600 ± 2081.0164	0.1437 ± 0.0328	0.1051 ± 0.0246
		BMF-CG-MIP(1)	28821.0000 ± 0.0000	0.0977 ± 0.0000	0.1322 ± 0.0000
		BMF-CG-MIP$_F$	26689.0000 ± 0.0000	0.1047 ± 0.0000	0.1429 ± 0.0000
		BMF-GA	15008.2000 ± 394.7454	0.3233 ± 0.0101	0.4379 ± 0.0139
		CNO-BMF (herein)	**65.2000 ± 3.2532**	**0.9896 ± 0.0016**	**0.9938 ± 0.0003**

partial simultaneous update mode. *IEEE Transactions on Neural Networks*, *10*, 975–978.

Leung, M.-F., & Wang, J. (2022). Cardinality-constrained portfolio selection based on collaborative neurodynamic optimization. *Neural Networks*, *145*, 68–79.

Leung, M.-F., Wang, J., & Che, H. (2022). Cardinality-constrained portfolio selection via two-timescale duplex neurodynamic optimization. *Neural Networks*, *153*, 399–410.

Li, T. (2005). A general model for clustering binary data. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 188–197).

Li, X., Wang, J., & Kwong, S. (2022). Boolean matrix factorization based on collaborative neurodynamic optimization with Boltzmann machines. *Neural Networks*, *153*, 142–151.

Liang, L., Zhu, K., & Lu, S. (2020). Bem: Mining coregulation patterns in transcriptomics via boolean matrix factorization. *Bioinformatics*, *36*, 4030–4037.

Likas, A., & Stafylopatis, A. (1996). Group updates and multiscaling: An efficient neural network approach to combinatorial optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *26*, 222–232.

Liu, N., Wang, J., & Qin, S. (2022). A one-layer recurrent neural network for nonsmooth pseudoconvex optimization with quasiconvex inequality and affine equality constraints. *Neural Networks*, *147*, 1–9.

Lu, H., Chen, X., Shi, J., Vaidya, J., Atluri, V., Hong, Y., & Huang, W. (2020). Algorithms and applications to weighted rank-one binary matrix factorization. *ACM Transactions on Management Information Systems*, *11*, 1–33.

Lu, H., Vaidya, J., & Atluri, V. (2008). Optimal Boolean matrix decomposition: Application to role engineering. In *2008 IEEE 24th International Conference on Data Engineering* (pp. 297–306). IEEE.

Lu, H., Vaidya, J., & Atluri, V. (2014). An optimization framework for role mining. *Journal of Computer Security*, *22*, 1–31.

Lucchese, C., Orlando, S., & Perego, R. (2010). Mining top-k patterns from binary datasets in presence of noise. In *Proceedings of 10th SIAM International Conference on Data Mining* (pp. 165–176).

Meeds, E., Ghahramani, Z., Neal, R., & Roweis, S. (2006). Modeling dyadic data with binary latent factors. *Advances in Neural Information Processing Systems*, *19*.

Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., & Mannila, H. (2008). The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, *20*, 1348–1362.

Muñoz-Pérez, J., Ruiz-Sepúlveda, A., & Benítez-Rochel, R. (2011). Parallelism in binary hopfield networks. In *11th International Work-Conference on Artificial Neural Networks* (pp. 105–112). Springer.

Ravanbakhsh, S., Póczos, B., & Greiner, R. (2016). Boolean matrix factorization and noisy completion via message passing. In *International Conference on Machine Learning* (pp. 945–954). PMLR.

Samaria, F. S., & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision* (pp. 138–142). IEEE.

Shen, B.-H., Ji, S., & Ye, J. (2009a). Mining discrete patterns via binary matrix factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 757–766).

Shen, B. H., Ji, S., & Ye, J. (2009b). Mining discrete patterns via binary matrix factorization. In *Knowledge Discovery and Data Mining*.

Slawski, M., Hein, M., & Lutsik, P. (2013). Matrix factorization with binary components. *Advances in Neural Information Processing Systems*, *26*.

Snášel, V., Platoš, J., & Krömer, P. (2008). On genetic algorithms for Boolean matrix factorization. In *2008 Eighth International Conference on Intelligent Systems Design and Applications* (pp. 170-175). IEEE volume 2.

Takefuji, Y., & Lee, K.-C. (1989). A near-optimum parallel planarization algorithm. *Science*, *245*, 1221–1223.

Takefuji, Y., & Lee, K. C. (1991). Artificial neural networks for four-coloring map problems and k-colorability problems. *IEEE Transactions on Circuits and Systems*, *38*, 326–333.

Wang, J., Wang, J., & Che, H. (2020). Task assignment for multivehicle systems based on collaborative neurodynamic optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *31*, 1145–1154.

Wang, J., Wang, J., & Han, Q.-L. (2021). Multi-vehicle task assignment based on collaborative neurodynamic optimization with discrete Hopfield networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*, 5274–5286.

Wei, L., Jin, L., & Luo, X. (2024). A robust coevolutionary neural-based optimization algorithm for constrained nonconvex optimization. *IEEE Transactions on Neural Networks and Learning Systems*, . In press.

Xia, Z., Liu, Y., & Wang, J. (2024). An event-triggered collaborative neurodynamic approach to distributed global optimization. *Neural Networks*, *169*, 181–190.

Xia, Z., Liu, Y., Wang, J., & Wang, J. (2023). Two-timescale recurrent neural networks for distributed minimax optimization. *Neural Networks*, *165*, 527–539.

Yan, Z., Fan, J., & Wang, J. (2017). A collective neurodynamic approach to constrained global optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *28*, 1206–1215.

Yan, Z., Wang, J., & Li, G. (2014). A collective neurodynamic optimization approach to bound-constrained nonconvex optimization. *Neural Networks*, *55*, 20–29.

Zhang, Y., Wang, S., Phillips, P., & Ji, G. (2014). Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowledge-Based Systems*, *64*, 22–31.

Zhang, Z., Li, T., Ding, C., & Zhang, X. (2007). Binary matrix factorization with applications. In *Seventh IEEE International Conference on Data Mining* (pp. 391–400).

Zhang, Z.-Y., Li, T., Ding, C., Ren, X.-W., & Zhang, X.-S. (2010). Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, *20*, 28–52.