Supplementary Material for "Estimating Dependency Structures for non-Gaussian Components with Linear and Energy Correlations"

Hiroaki Sasaki[1] hsasaki@cc.uec.ac.jp
Michael U. Gutmann[2] michael.gutmann@helsinki.fi
Hayaru Shouno[1] shouno@uec.ac.jp
Aapo Hyvärinen[3,4] aapo.hyvarinen@helsinki.fi

[1]Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan
[2]Department of Mathematics and Statistics and HIIT, University of Helsinki, Helsinki, Finland
[3]Department of Computer Science and HIIT, University of Helsinki, Helsinki Finland
[4] ATR Cognitive Mechanisms Laboratories, Kyoto, Japan

A Details for the Permutation Algorithm

Here, we describe the permutation algorithm used in Section 3. The goal of this algorithm is to estimate an order vector $\boldsymbol{k} = (k_1, k_2, \ldots, k_d)$ where $k_i \in \{1, 2, \ldots, d\}$ so that $\mathbf{M}$ in Figure 2(b) approaches a tridiagonal matrix. The algorithm is as follows:

Greedy Permutation Algorithm

Input: $\mathbf{W}$ and $\mathbf{M}$.

- Initialization: Set $\hat{k}_1 = 1$ and the remaining index set $\mathbf{I} = \{2, \ldots, d\}$, and make a matrix with zero diagonal elements by $\mathbf{M}' = \mathbf{M} - \mathbf{M}_{diag}$ where $\mathbf{M}_{diag}$ denotes the diagonal matrix whose diagonal elements are the diagonal ones in $\mathbf{M}$.
- Repeat the following procedures from $i = 2$ to $i = d$:
  - Find $\hat{k}_i$ by
  $$\hat{k}_i = \arg\max_{j \in \mathbf{I}} m'_{\hat{k}_{i-1}, j} \quad \text{(A1)}$$
  where $m'_{\hat{k}_{i-1}, j}$ denotes the $(\hat{k}_{i-1}, j)$-th element in $\mathbf{M}'$.
  - Update $\mathbf{I}$ by removing $\hat{k}_i$:
  $$\mathbf{I} \leftarrow \mathbf{I} \setminus \hat{k}_i. \quad \text{(A2)}$$

Output: $\hat{\boldsymbol{k}} = (\hat{k}_1, \hat{k}_2, \ldots, \hat{k}_d)$.

By (A1), $m_{\hat{k}_{i-1}, \hat{k}_i}$ for all i are made to take a large value, and thus, $\hat{M}$, whose (i, j)-th element is $m_{\hat{k}_i, \hat{k}_j}$, should be close to a tridiagonal matrix. Using $\hat{\boldsymbol{k}}$, the row vectors in $\mathbf{W}$ were permuted as $\mathbf{W}_p = (\boldsymbol{w}_{\hat{k}_1}, \boldsymbol{w}_{\hat{k}_2}, \ldots, \boldsymbol{w}_{\hat{k}_d})^\top$ where $\boldsymbol{w}_i$ denotes the i-th row vector in $\mathbf{W}$, and in Figure 2(c) and (d), we visualized the performance matrix as $\mathbf{W}_p\mathbf{A}$ and correlation matrix for the permuted sources $\mathbf{W}_p\boldsymbol{x}$.

B Undirected Graph for Natural Images

This section presents an undirected graph for natural images as in Figure 6 for the outputs of simulated complex cells. The graph is depicted in Figure A. It shows that features with similar orientations or positions tend to be (conditionally) dependent.

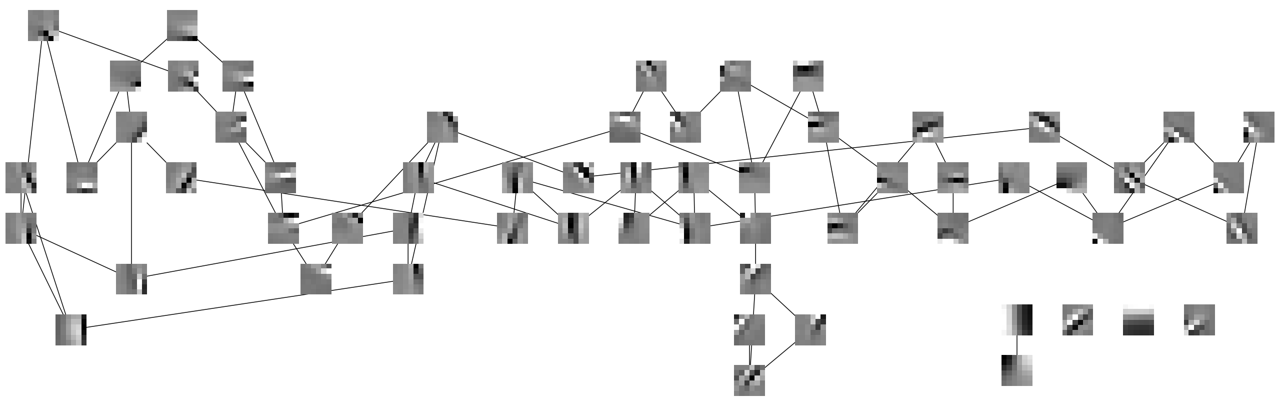

Figure A: An undirected graph for natural images. Each node corresponds to a feature estimated from natural images, and only the links which have $m_{i,j}$ larger than 0.8 are displayed.