Supplemental Material for

Adaptive Cluster Expansion for Boltzmann Machines with Noisy Data

S. Cocco, R. Monasson

This document details the pseudo-codes useful for the practical implementation of the inference algorithm. Given a cluster Γ the partition function of the K–spin system restricted to Γ,

$$Z(\mathbf{J},\Gamma) = \sum_{\{\sigma_i=0,1;\ i\in\Gamma\}} \exp\Big(\sum_{i\in\Gamma} h_i\sigma_i + \sum_{i<j;\ i,j\in\Gamma} J_{ij}\,\sigma_i\sigma_j\Big)\ , \tag{1}$$

can be computed in time $\propto 2^K$. The entropy $S(\Gamma)$ is then obtained from definition (2) (main text) and the use of a convex minimization algorithm. The computation of the entropy S_0 requires the eigenvalues Λ_a of the matrix $\hat{\mathbf{c}}(\Gamma)$, the $K\times K$ restriction of the matrix $\hat{\mathbf{c}}$ to the spins in Γ:

$$S_0(\Gamma) = \frac{1}{2}\sum_{a=1}^{K} \log\Lambda_a\ . \tag{2}$$

In the case of undersampling, some eigenvalues Λ_a may be equal to zero. To regularize the expressions of S and S_0 we introduce a L_2 penalty $\mu\sum_{i<j} J_{ij}^2\, p_i(1-p_i)p_j(1-p_j)$ over the couplings, where $\mu>0$. Formula (2) above becomes

$$S_0(\Gamma) = \frac{1}{2}\sum_{a=1}^{K} \big(\log\hat\Lambda_a + 1 - \hat\Lambda_a\big)\ , \tag{3}$$

where

$$\hat\Lambda_a = \frac{1}{2}\left(\Lambda_a - \mu + \sqrt{(\Lambda_a-\mu)^2+4\mu}\right) \tag{4}$$

is strictly positive. The optimal value for μ can be determined through Bayesian inference, see D.J.C. MacKay, *Neural Computation* **4**, 415 (1991).

The first pseudo-code obtains $\Delta S(\Gamma)$ (through Möbius inversion formula):

Algorithm 1 Computation of cluster-entropy $\Delta S(\Gamma)$

Require: Γ (of size K), $\mathbf{p}$, routines to calculate S_0 and S
$\Delta S(\Gamma) \leftarrow S(\Gamma) - S_0(\Gamma)$
SIGN $\leftarrow 1$
for SIZE $= K-1$ **to** 1 **do**
SIGN $\leftarrow -$ SIGN
for every Γ' with SIZE spins in Γ **do**
$\Delta S(\Gamma) \leftarrow \Delta S(\Gamma)+$ SIGN $\times\big(S(\Gamma') - S_0(\Gamma')\big)$
end for
end for
Output: $\Delta S(\Gamma)$

When the above routine is called several times (to compute the entropies of various clusters) a substantial speed-up can be achieved by memorizing the entropies $\Delta S(\Gamma)$ of every cluster. The above routine is simply turned into a recursive procedure by changing the line $\Delta S(\Gamma) \leftarrow \Delta S(\Gamma)+$ SIGN $\times\big(S(\Gamma') - S_0(\Gamma')\big)$ into $\Delta S(\Gamma) \leftarrow \Delta S(\Gamma) - \Delta S(\Gamma')$ – the variable SIGN becomes useless. Hence, the computation of $S(\Gamma)$, which is the slowest step for large cluster sizes, is done only once for every Γ.

The core of our inference algorithm is the recursive building-up and the selection of new clusters:

Algorithm 2 Adaptive Cluster Expansion

Require: N, Θ, S_0, routine to calculate $\Delta S(\Gamma)$ from $\mathbf{p}$
LIST $\leftarrow \emptyset$ {All selected clusters}
SIZE $\leftarrow 1$
LIST(1) $\leftarrow (1) \cup (2) \cup \ldots \cup (N)$ {Clusters of SIZE=1}
repeat {Building-up of clusters with one more spin}
LIST $\leftarrow$ LIST $\cup$ LIST(SIZE) {Store current clusters}
LIST(SIZE+1) $\leftarrow \emptyset$
for every pairs $\Gamma_1, \Gamma_2 \in$ LIST(SIZE) **do**
$\Gamma_I \leftarrow \Gamma_1 \cap \Gamma_2$ {Spins belonging to Γ_1 and to Γ_2}
$\Gamma_U \leftarrow \Gamma_1 \cup \Gamma_2$ {Spins belonging to Γ_1 or to Γ_2}
if Γ_I contains (SIZE-1) spins **and** $|\Delta S(\Gamma_U)| > \Theta$ **then**
LIST(SIZE+1) $\leftarrow$ LIST(SIZE+1) $\cup\ \Gamma_U$ {add Γ_U to the list of selected clusters}
end if
end for
SIZE $\leftarrow$ SIZE+1
until LIST(SIZE) $= \emptyset$
$S \leftarrow S_0$, $J \leftarrow -\frac{d}{d\mathbf{p}} S_0$ {Calculation of S, $\mathbf{J}$}
for $\Gamma \in$ LIST **do**
$S \leftarrow S + \Delta S(\Gamma)$, $\mathbf{J} \leftarrow \mathbf{J} - \frac{d}{d\mathbf{p}} \Delta S(\Gamma)$
end for
Output: S, $\mathbf{J}$ and LIST of clusters