December 1993
LU TP 93-28
Revised

Clustering Noisy Data with Deterministic Annealing

Mattias Ohlsson[1]
Department of Theoretical Physics
University of Lund
Sölvegatan 14A
S-223 62 Lund, Sweden

Submitted to **Neural Computation**

Abstract:
A cluster algorithm for noisy data distributions is presented. It minimizes an error function using a deterministic annealing procedure. Phase transitions occur during the annealing as large clusters split into smaller ones. Critical "temperatures" corresponding to these transitions are estimated in order to make the annealing as efficient as possible. The approach is successfully tested on data sets containing up to 10 clusters contaminated with 100% noise.

[1] mattias@thep.lu.se

1 Introduction

An important problem in pattern recognition and computer vision is the detection of clusters in the presence of noise. From a given observed data distribution one should replace all data points by a set of representative vectors or cluster centers such that the information loss is minimized. Very often, none or very limited a priori information about the distribution is available. Also, in many real-world applications the observed distribution can be very noisy, in which case some data points should not be assigned to any cluster at all. If the underlying distribution has a known parametric form, standard methods like Bayesian learning [1] can be applied. However, in cases where no such prior information is available alternative non-parametric methods must be used. One class of such methods are algorithms that minimize a given error function (or distortion measure). The well-known k-means clustering algorithm [2] is such an example.

The algorithm derived in this paper minimizes an error function that is suitable for data distributions containing a lot of noise. It is inspired by a method used for track finding in high energy physics [3, 5] where a set of deformable templates are adjusted to match real tracks represented by the data set. The cluster problem can be viewed as a special case of the track finding problem where each deformable template now represents possible cluster centers matching the data set. The algorithm converges using a deterministic annealing procedure, which corresponds to minimizing the free energy of a Boltzmann distribution of the error function. Related algorithms using statistical mechanics as a tool for the optimization procedure in clustering are found in refs. [6, 7, 8, 9, 10].

The main results in our approach are:

- The error function allows for unassigned data points thereby neglecting data points corresponding to noise. The amount of noise points the algorithm allows for is governed by the parameter λ which can be interpreted as the square of a width of a *zero neuron* collecting noise points.

- Phase transition properties of the algorithm are discussed and critical temperatures are derived for clusters splitting in a hierarchical way.

- The solution quality is not sensitive to the number of initial clusters K.

- Numerical studies on simulated data shows good solution quality for problem sizes up to 5000 data points with a 100% noise level.

2 The Algorithm

The problem is to replace a data set $\{\mathbf{x}_i | i = 1, ... N\}$ by a set of representative vectors $\{\mathbf{y}_a | a = 1, ..., K\}$, such that the information loss is minimized. If the data contains no noise the above objective can be achieved by minimizing the distortion error

$$E'(\mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \min_{a} (M_{ia}), \quad (a = 1, ..., K) \ , \tag{1}$$

where $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_K)$ and M_{ia} is a distortion measure between data point $\mathbf{x}_i$ and cluster center $\mathbf{y}_a$. There are many possible choices for M_{ia} [9]. Because of simplicity M_{ia} is in what follows taken to be the half squared Euclidean distance between $\mathbf{x}_i$ and $\mathbf{y}_a$,

$$M_{ia} = \frac{1}{2} |\mathbf{x}_i - \mathbf{y}_a|^2 \ . \tag{2}$$

On the other hand, if the data contains noise that should be ignored, then E' is modified according to [3, 4, 5]

$$E(\{S_{ia}\}, \mathbf{Y}) = \sum_{i=1}^{N} \sum_{a=1}^{K} S_{ia} M_{ia} + \lambda \sum_{i=1}^{N} \left(\sum_{a=1}^{K} S_{ia} - 1 \right)^2 \ , \tag{3}$$

where S_{ia} is a logical decision unit such that $S_{ia} = 1$ if data point i is assigned to cluster a and zero otherwise. The parameter λ imposes a penalty if point i is not assigned to any cluster center a. In order to allow for possible noise points the matrix S, with matrix elements S_{ia}, is subject to the constraint

$$\sum_{a=1}^{K} S_{ia} = 1 \text{ or } 0 \quad \forall \, i \ . \tag{4}$$

In this way the parameter λ governs the amount of noise the algorithm allows for.

When minimizing E in eq. (3) subject to the constraint of eq. (4), fluctuations are introduced into the system using *simulated annealing* [11] where the system is allowed to thermalize for a sequence of temperatures $T_n > T_{n-1} > ... > T_0$ according to the Boltzmann distribution

$$P(\{S_{ia}\}, \mathbf{Y}) = \frac{1}{Z} e^{-\beta E(\{S_{ia}\}, \mathbf{Y})} \ , \tag{5}$$

where $\beta = 1/T$ and Z is a normalization constant, the so-called partition function.

In order to obtain the marginal probability distribution $P_M(\mathbf{Y})$ we must sum $P(\{S_{ia}\}, \mathbf{Y})$ over all matrices S that satisfies the constraint eq. (4). Doing this (for details see appendix A) we end up with

$$P_M(\mathbf{Y}) = \frac{1}{Z} e^{-\beta E_{\mathit{eff}}(\mathbf{Y})} \ , \tag{6}$$

where the *effective* error E_{eff} is introduced as

$$E_{\mathit{eff}}(\mathbf{Y}) = -\frac{1}{\beta}\sum_{i=1}^{N}\log\left(e^{-\beta\lambda} + \sum_{a=1}^{K} e^{-\beta M_{ia}}\right) \ . \tag{7}$$

At a given temperature $(1/\beta)$, the most probable configuration according to eq. (6) is given by the minima of E_{eff}. Using a gradient descent method to minimize E_{eff} one gets the updating rule

$$\begin{aligned} \mathbf{y}_a \quad &\rightarrow \quad \mathbf{f}_a(\mathbf{Y}) \equiv \mathbf{y}_a - \epsilon\nabla_a E_{\mathit{eff}} = \\ &= \quad \mathbf{y}_a + \epsilon\sum_i V_{ia}(\mathbf{x}_i - \mathbf{y}_a) \ , \end{aligned} \tag{8}$$

where ϵ is the step size and the *Potts neuron* V_{ia} is given by

$$V_{ia} = \frac{e^{-\frac{1}{2}\beta|\mathbf{x}_i - \mathbf{y}_a|^2}}{e^{-\beta\lambda} + \sum_b e^{-\frac{1}{2}\beta|\mathbf{x}_i - \mathbf{y}_b|^2}} \ . \tag{9}$$

Eq. (8) is easy to interpret. Each cluster center $\mathbf{y}_a$ takes small steps towards signal points $\mathbf{x}_i$ with a relative strength given by V_{ia}. The latter is bounded by $0 < V_{ia} < 1$ and has the natural interpretation, $V_{ia} = < S_{ia} >_\beta$, as the thermal average of the binary assignment variable S_{ia}.

The role played by the parameter λ can be seen by rewriting V_{ia} as

$$V_{ia} = \frac{1}{e^{-\beta(\lambda - M_{ia})} + \sum_b e^{-\beta(M_{ib} - M_{ia})}} \ . \tag{10}$$

Figure 1 shows V_{ia} as a function of M_{ia} for three different values of β and for the special case of $\lambda = 1$ and $M_{ib} = M_{ia}$ ($\forall\ b$). If $M_{ia} < \lambda$ then $V_{ia} \rightarrow 1/K$ as $\beta \rightarrow \infty$, on the other hand if $M_{ia} > \lambda$ then $V_{ia} \rightarrow 0$ as $\beta \rightarrow \infty$. In general we can interpret λ as a border for noise rejection. This border is fuzzy for small β, while for large β there is a sharp transition from $1/K$ to 0 (see fig. (1)). In the large β limit only data points within a distance measure λ from any $\mathbf{y}_a$ are important when updating the cluster centers.

The well known K-means clustering algorithm [2] corresponds to $\lambda = \infty$, $\beta = \infty$ and the number of initial cluster centers K, fixed a priori.

The parameter λ makes it natural to introduce, at least formally, the notation of a *zero neuron* V_{i0},

$$V_{i0} = \frac{e^{-\beta\lambda}}{e^{-\beta\lambda} + \sum_b e^{-\beta M_{ib}}} \ . \tag{11}$$

The zero neuron consists of all data points having $\min_a(M_{ia}) \geq \lambda$. Since, $\sum_{a=0}^{K} V_{ia} = 1$, we can interpret V_{ia} as the probability, at a given β, for data point i to belong to cluster a. A

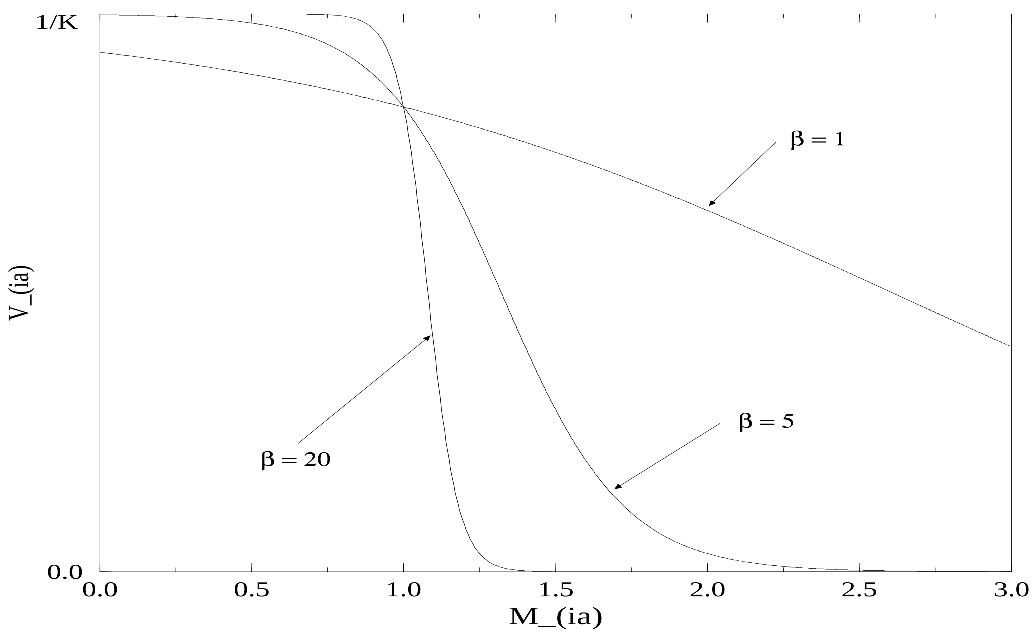

Figure 1: This figure shows V_{ia} as a function of M_{ia} for $\beta = 1, 5, 20$ together with $\lambda = 5$ and $M_{ib} = M_{ia}$ $(\forall\ b)$. V_{ia} varies slowly for small β making a fuzzy border for the noise rejection. For large values of β however, there is a sharp transition from $1/K$ to 0 as M_{ia} becomes larger than λ.

cluster probability $P_\beta(a)$ can be defined as

$$P_\beta(a) = \frac{1}{N} \sum_{i=1}^{N} V_{ia} \quad , \tag{12}$$

which gives the probability for $\mathbf{y}_a$ being a cluster center. $P_\beta(a)$ can be used after convergence of the algorithm to delete non-valid clusters having a small $P_\beta(a)$.

Using cluster probabilities as a regularization tool is used in ref. [9] by including $P_\beta(a)$ in the error measure E

$$E = \sum_{i,a} S_{ia} \left[M_{ia} + \gamma C_a(P_a) \right] \quad , \tag{13}$$

where C_a is function of the cluster probabilities P_a. Any a priori information about the clustering problem is then put into the functional form of the C_a's.

3 Dynamical Properties

The update equation (eq. (8)) above determines, for a fixed β, a set of cluster centers $\{\mathbf{y}_a | a = 1, ..., K\}$. For $\beta = 0$ ($T = \infty$) every configuration becomes equally probable and all $\mathbf{y}_a$'s will merge into a common fix point $\mathbf{y}_*$ located at the center of mass[2]. For nonzero, but small β $\mathbf{y}_*$ is implicitly given by

$$\mathbf{y}_* = \frac{\sum_i V_{i*} \mathbf{x}_i}{\sum_i V_{i*}} \quad \approx \quad -\frac{\beta}{(K+1)} \frac{1}{N} \sum_i \mathbf{x}_i^2 \mathbf{x}_i \quad , \tag{14}$$

where V_{i*} denotes the the Potts factor evaluated at the fixpoint $\mathbf{y}_*$. At this point only one cluster occur - the entire data set.

If β is increased further, then at some $\beta = \beta_c$ the fixpoint $\mathbf{y}_*$ will become unstable and the single cluster splits into one or more new clusters as the system undergoes a phase transition [6].

For synchronous updating β_c can be found by studying the eigenvalues to the Jacobian matrix M, defined by

$$\mathsf{M}_{ab} = \frac{\partial}{\partial \mathbf{y}_b} \otimes \mathbf{f}_a \quad , \tag{15}$$

$$\frac{\partial}{\partial \mathbf{y}_b} \equiv \left(\frac{\partial}{\partial \mathbf{y}_{b1}}, ..., \frac{\partial}{\partial \mathbf{y}_{bD}} \right) \quad , \tag{16}$$

where $\mathbf{f}_a$ is defined in eq. (8) and D is the dimension. It is clear that $\mathbf{y}_*$ will become unstable when the absolute value of any eigenvalue of M grows larger than unity. Straight forward calculation gives

$$\mathsf{M}_{ab} = \delta_{ab} \mathsf{D} + \mathsf{B} \quad , \tag{17}$$

where δ_{ab} is the Kronecker δ-symbol and the matrices D and B are given by

$$\mathsf{D} = \mathsf{I}^{(D)} \left(1 - \epsilon \sum_i V_{i*} \right) + \epsilon\beta \sum_i V_{i*} (\mathbf{x}_i - \mathbf{y}_*) \otimes (\mathbf{x}_i - \mathbf{y}_*) \tag{18}$$

$$\mathsf{B} = -\epsilon\beta \sum_i V_{i*}^2 (\mathbf{x}_i - \mathbf{y}_*) \otimes (\mathbf{x}_i - \mathbf{y}_*) \quad , \tag{19}$$

and $\mathsf{I}^{(D)}$ denotes the D-dimensional identity matrix. M can now be written as

$$\mathsf{M} = \mathsf{D} \otimes \mathsf{I}^{(K)} + \mathsf{B} \otimes K\mathsf{P} \quad , \tag{20}$$

with $\mathsf{P}_{ab} = 1/K$, $(a, b = 1, ..., K)$. From eq. (20) it follows that two different modes of splitting occur, either in the parallel mode where the eigenvector $\mathbf{Y}$ satisfies $\mathsf{P}\mathbf{Y} = \mathbf{Y}$, or

[2] center of mass is, without loss of generality, taken to be the origin.

in the transverse mode with $\mathsf{P}\mathbf{Y} = \mathbf{0}$. In the latter all clusters will move away from the fixpoint with their center of mass conserved and this is experimentally always the case. For the transverse mode where the eigenvalues of M are given by the eigenvalues α_i of D, one has

$$\alpha_{max/min} = 1 - \epsilon \sum_i V_{i*} + \epsilon\beta\gamma_{max/min} \quad . \tag{21}$$

In eq. (21) $\gamma_{max/min}$ is the largest/smallest eigenvalue of the matrix $\sum_i V_{i*}(\mathbf{x}_i - \mathbf{y}_*) \otimes (\mathbf{x}_i - \mathbf{y}_*)$. The fixpoint becomes unstable if $(\alpha_{max} > 1)$ or $(\alpha_{min} < -1)$. The latter, however, cannot occur as long as $\epsilon \leq K/N$. The condition for instability is therefore $\alpha_{max} = 1$, which gives β_c implicitly from

$$\beta_c = \frac{\sum_i V_{i*}}{\gamma_{max}} \quad . \tag{22}$$

For serial updating the analysis becomes somewhat more complicated. The idea is to bring it back to the synchronous case using an effective updating matrix (for details see appendix B).

So far only the splitting of the first cluster has been analyzed. Critical temperatures for subsequent splittings can be found approximately using the method above, but only with the subset of data points closest to the splitting cluster.

For the cluster problems studied in this paper β_c is much larger than the average M_{ia}, therefore the $\beta = 20$ curve in figure 1 is the relevant picture for V_{ia} when the first splitting occur. In order to take into account all possible clusters λ should initially be set to a relatively high value. As more and more clusters are being formed λ must be decreased in order to reject possible noise data points.

4 Simulations and Results

4.1 Implementation Issues

This algorithm can be implemented in different ways depending on the known a priori information about the clustering problem. In its "raw" form it contains two parameters K, the number of initial cluster centers, and the noise parameter λ. In all of the problems studied in this paper good solutions were found having a large initial λ and gradually decreasing λ to a small value at the point of convergence. This leaves only K as a parameter that has to be set by the user. The solution quality is, however, not sensitive to K as long as K is larger than or equal to the true number of clusters. If necessary, degenerate $\mathbf{y}_a$ can be removed afterwards by a simple heuristic. Figure 2 shows the implementation scheme used for the numerical simulations in this paper. No effort was made on how to find an

1. Rescale the data distribution to a predefined dynamical range R, such that $|\mathbf{x}_i| \in [-R, R]$ (we used $R = 1$). Transform data to the center of mass system $\sum_i \mathbf{x}_i = 0$.
2. Choose K and set $\lambda_{start} = R^2$, since initially all data points are important. Set the update parameter ϵ in eq. (8) to the value K/N
3. Find the critical β_c as described in the above section.
4. Start the annealing procedure by setting $\beta = 0.95 * \beta_c$, $\lambda = \lambda_{start}$ and $\mathbf{y}_a = \mathbf{y}_* +$ "small random vector" $(a = 1, ..., K)$.
5. Update cluster centers $\mathbf{y}_a$ according to eq. (8). Both synchronous and serial updating is possible. We use synchronous updating since it is more rapid.
6. Let $\lambda \to 0.98 * \lambda$ and $\beta \to 1.01 * \beta$.
7. `If` $\beta < \beta_{max}$ goto 5 `else` continue.
8. Delete degenerate clusters, that is, reject $\mathbf{y}_a$ if $\mathbf{y}_a \equiv \mathbf{y}_b$ for $a \neq b$.
9. Reject clusters with a small P_β (see eq. (12)).

Figure 2: The cluster algorithm with a deterministic annealing procedure suitable for data sets contaminated with noise.

optimal β_{max} for each individual problem and we simply used a fixed $\beta_{max} = 1200$ for the problems studied.

4.2 Numerical Tests

The algorithm is tested on different problems using the k-means algorithm as a reference. The performance of either method is measured by Δ, given by

$$\Delta = \frac{1}{K'\bar{D}} \max \left[\sum_{a=1}^{K} \min_b (D_{ab}) \ , \ \sum_{b=1}^{K'} \min_a (D_{ab}) \right] \ , \tag{23}$$

where K' is the correct number of clusters and D_{ab} is the distance between $\mathbf{y}_a$ and cluster b. $\bar{D}$ denotes the average distance between true clusters and is used as a normalization. A small Δ indicates a good solution while a large Δ means that either too many or too few clusters were found.

The data sets consists of clusters generated by normally distributed sources, with equal width, such that each cluster has a fractional population that lies within $[0.3, 1]$. There

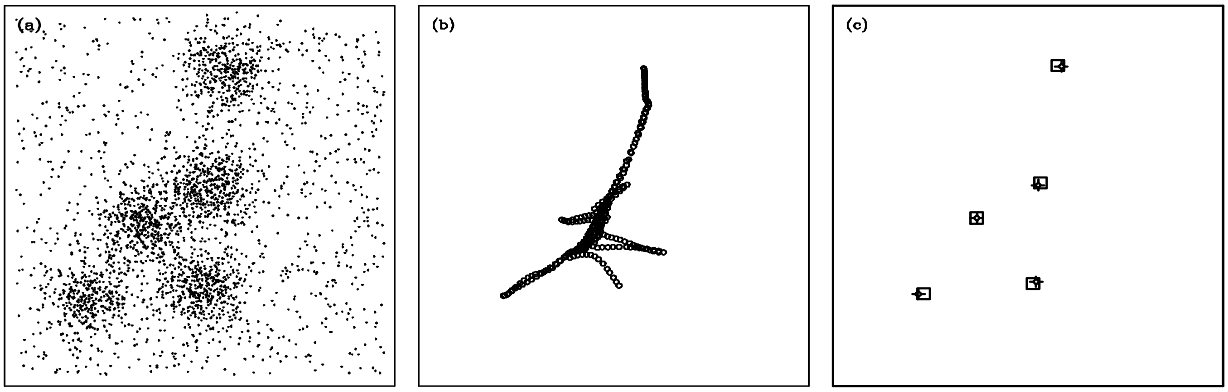

Figure 3: **(a)** Data set with 3000 points, 5 clusters and a noise level of 50%. **(b)** Development of the individual $\mathbf{y}_a$ during the annealing procedure. **(c)** The true cluster locations (stars) and the solution found after convergence (squares).

Algorithm	$N = 2000$ noise $= 0\%$ # clusters $= 4$		$N = 3000$ noise $= 50\%$ # clusters $= 4$		$N = 3000$ noise $= 50\%$ # clusters $= 6$		$N = 5000$ noise $= 100\%$ # clusters $= 10$	
	K	Δ	K	Δ	K	Δ	K	Δ
Our method	4	0.07 ± 0.07	4	0.05 ± 0.08	6	0.09 ± 0.07	10	0.09 ± 0.04
Our method	8	0.10 ± 0.09	8	0.04 ± 0.05	10	0.08 ± 0.10	15	0.10 ± 0.05
Our method	12	0.13 ± 0.15	12	0.10 ± 0.18	14	0.07 ± 0.08	20	0.11 ± 0.06
K-means	4	0.09 ± 0.07	4	0.31 ± 0.19	6	0.28 ± 0.13	10	0.24 ± 0.08
K-means	8	0.37 ± 0.18	8	0.87 ± 0.32	10	0.53 ± 0.14	15	0.42 ± 0.08

Table 1: Comparisons of performance for the deterministic annealing method and the k-means clustering algorithm. Δ is an average taken over 50 independent runs.

is no constraint for the cluster locations and it may happen that clusters overlap heavily. Figure 3 shows a cluster problem with 5 clusters and a 50% noise level ($N = 3000$) together with the development of the individual $\mathbf{y}_a$ during annealing ($K = 10$). It is encouraging to see how the algorithm finds the correct solution despite the high noise level (see fig. 3c).

Table 1 summarizes the comparison between k-means and the method of figure 2. Each Δ shown is an average over 50 independent runs. The rather large variance for our method originates from a very few solutions where too many clusters are found and where the simple P_β-criteria (see eq. (12)) failed to remove superfluous $\mathbf{y}_a$. In any case it outperforms the k-means method even when the latter is always initialized with the correct number of clusters.

5 Conclusion

We have devised a cluster finding method suitable for data distributions contaminated with a lot of noise. It converges using a deterministic annealing procedure.

Phase transition temperatures where the initial clusters split are derived and used to initiate the algorithm thereby avoiding unnecessary CPU consumption.

The algorithm is related to robust statistics - it ignores noise to a desired level. The approach is easy to adapt to specific situations. For example, suppose each data point comes with a measure of "goodness". Then the formalism can be generalized to allow for i-dependent λ's ($\lambda \rightarrow \lambda_i$).

The specific implementation of our approach used in this paper has one external parameter K - the number of initial cluster centers. However, one only has to estimate the upper bound for the number of clusters present in data sets one studies.

The approach has been tested on different clustering problems with encouraging results. The good performance is possible because the formalism allows for possible unassigned data points thereby making it different from other related approaches [6, 9, 10].

The approach scales like NxK and has the advantage of being intrinsically parallel.

Acknowledgements

This work was supported by the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine.

Appendix A

This appendix contains a calculation of the marginal probability distribution $P_M(\mathbf{Y})$. We start from the error measure

$$E(\{S_{ia}\}, \mathbf{Y}) = \sum_{i,a} S_{ia} M_{ia} + \lambda \sum_i \left(\sum_a S_{ia} - 1 \right)^2 , \tag{A1}$$

with $i = 1, ..., N$ and $a = 1, ..., K$. Define $\sigma(i) = \sum_a S_{ia}$, then the constraint for S is given by

$$\sigma(i) = 1 \text{ or } 0 \quad \forall\, i\ . \tag{A2}$$

We next calculate the marginal probability distribution $P_M(\mathbf{Y})$ given by

$$P_M(\mathbf{Y}) = \sum_{\{\mathsf{S}\}} P(\{S_{ia}\}, \mathbf{Y}) \ \ , \tag{A3}$$

where the sum is over all matrices S satisfying eq. (A2). There are $(K+1)^N$ such different matrices. Using eq. (A1) we get

$$ZP_M = Z \sum_{\{\mathsf{S}\}} P = \sum_{\{\mathsf{S}\}} e^{-\beta \sum_i \left(\sum_a S_{ia} M_{ia} + \lambda\left(\sum_a S_{ia} - 1\right)^2\right)}\ . \tag{A4}$$

Now replace $\sum_{\{\mathsf{S}\}}$ by $\sum_{\{\alpha(i)\}}$, where $\alpha(i)$ is defined as follows:
For any matrix S satisfying the constraint one has

$$S_{i\alpha(i)} = 1 \ \text{ if } \ \sigma(i) = 1\ . \tag{A5}$$

If $\sigma(i) = 0$ we set $\alpha(i) = 0$ since in any case $\sum_a S_{ia} = 0$.

The marginal probability can now be written as

$$\begin{aligned} ZP_M &= \sum_{\{\alpha(i)\}} \exp\left(-\beta \sum_i \left[M_{i\alpha(i)}\delta_{\sigma(i),1} + \lambda \delta_{\sigma(i),0}\right]\right) = \\ &= \sum_{\{\alpha(i)\}} \prod_i \exp\left(-\beta\left[M_{i\alpha(i)}\delta_{\sigma(i),1} + \lambda\delta_{\sigma(i),0}\right]\right)\ \ , \end{aligned} \tag{A6}$$

where δ is the Kronecker delta symbol. This sum of $(K+1)^N$ products can, since the order of the different $\alpha(i)$'s are unimportant, be written as a N product of $(K+1)$ terms

$$ZP_M = \prod_i \left(e^{-\beta\lambda} + \sum_a e^{-\beta M_{ia}}\right)\ . \tag{A7}$$

It is now easy to write P_M as a Boltzmann distribution

$$P_M = \frac{1}{Z} e^{-\beta E_{\mathit{eff}}} \ \ , \tag{A8}$$

where the *effective* error E_{eff} is given by

$$E_{\mathit{eff}} = -\frac{1}{\beta} \sum_{i=1}^{N} \log\left(e^{-\beta\lambda} + \sum_{a=1}^{K} e^{-\beta M_{ia}}\right)\ . \tag{A9}$$

Appendix B

In this appendix we derive the effective synchronous updating matrix for serial updating. Start by expanding $\mathbf{f}_a$ in eq. (8) to the first order around the fixpoint $\mathbf{y}_*$. One gets

$$\begin{aligned} \mathbf{y}'_a &= \mathbf{f}_a(\mathbf{Y}) && \text{(A1)} \\ &= \mathsf{I}^{(D)}\mathbf{f}_a(\mathbf{Y}_*) + \sum_{b=1}^{K} \left(\frac{\partial}{\partial \mathbf{y}_b} \otimes \mathbf{f}_a \right) (\mathbf{y}_b - \mathbf{y}_*) \quad . && \text{(A2)} \end{aligned}$$

Straight forward calculations gives

$$\mathbf{y}'_a = \mathsf{D}\mathbf{y}_a + \sum_{b=1}^{K} \mathsf{B}\mathbf{y}_b - (\mathsf{D} + K\mathsf{B})\,\mathbf{y}_* \quad , \tag{A3}$$

where the matrices D and B are defined in eq. (18). Considering only the fluctuations $\mathbf{s}_a$ around the fixpoint the serial updating reads

$$\mathbf{s}'_a = (\mathsf{D} + \mathsf{B})\,\mathbf{s}_a + \sum_{b=1}^{a-1} \mathsf{B}\mathbf{s}'_b + \sum_{b=a+1}^{K} \mathsf{B}\mathbf{s}_b \quad , \tag{A4}$$

with the prime denoting the updated variable. Let $\mathbf{S}$ be the vector of all fluctuations, then

$$\mathbf{S}' = \left[(\mathsf{D} + \mathsf{B}) \otimes \mathsf{I}^{(K)} \right] \mathbf{S} + (\mathsf{B} \otimes \mathsf{L})\,\mathbf{S}' + (\mathsf{B} \otimes \mathsf{U})\,\mathbf{S} \quad . \tag{A5}$$

In eq. (A5) L and U are defined as,

$$\mathsf{L}_{ab} = \begin{cases} 1 & \text{if } a > b \\ 0 & \text{if } a \leq b \end{cases} \tag{A6}$$

$$\mathsf{U}_{ab} = \begin{cases} 1 & \text{if } a < b \\ 0 & \text{if } a \geq b \end{cases} \tag{A7}$$

Rearranging the terms in eq. (A5) we finally get,

$$\mathbf{S}' = \left(\mathsf{I}^{(KD)} - \mathsf{B} \otimes \mathsf{L} \right)^{-1} \left(\mathsf{B} \otimes \mathsf{U} + (\mathsf{D} + \mathsf{B}) \otimes \mathsf{I}^{(K)} \right) \mathbf{S} \quad . \tag{A8}$$

The matrix appearing in front of $\mathbf{S}$ can now be regarded as the effective synchronous updating matrix that corresponds to serial updating. The fixpoint $\mathbf{y}_*$ will become unstable when the absolute value of any eigenvalue of this matrix grows larger than unity.

References

[1] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York (1973).

[2] J. MacQueen, "Some methods for classification of and analysis of multivariate observations", *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, (1967) 281-297.

[3] A. Yuille, K. Honda and C. Peterson, “Particle Tracking by Deformable Templates”, *Proceedings of 1991 IEEE INNS International Joint Conference on Neural Networks* **1**, (1991) 7-12.

[4] M. Ohlsson, C. Peterson and A.L. Yuille, “Track finding with deformable templates - the elastic arms approach”, *Comput. Phys. Commun.* **71**, (1992) 77-98.

[5] M. Ohlsson, “Extensions and explorations of the elastic arms algorithm”, *Comput. Phys. Commun.* **77**, (1993) 19-32.

[6] K. Rose, E. Gurewitz and G.C. Fox “Statistical Mechanics and Phase Transitions in Clustering”, *Phys. Rev. Lett.* **65**, (1990) 945-948.

[7] K. Rose, E. Gurewitz and G.C. Fox “Vector Quantization by Deterministic Annealing”, *IEEE Trans. Inform. Theory* **38**, (1992) 1249-1257.

[8] K. Rose, E. Gurewitz and G.C. Fox “Constrained Clustering as an Optimization Method”, *IEEE Trans. Patt. Anal. Machine Intell.* **15**, (1993) 785-794.

[9] J. Buhmann and H. Kühnel, “Complexity Optimized Data Clustering by Competitive Neural Networks”, *Neural Comput.* **5**, (1993) 75-78.

[10] Y. Wong, “Clustering Data by Melting”, *Neural Comput.* **5**, (1993) 89-104.

[11] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, “Optimization by Simulated Annealing”, *Science* **220**, (1983) 671.