DA-GTM

Jong Youl Choi (jychoi@cs.indiana.edu)

1 Introduction

The Generative Topographic Mapping (GTM), also known as a principled alternative to the Self-Organizing Map (SOM), has been developed for modeling the probability density of data and its visualization in a lower dimension. Contrast to the SOM which does not define a density model [1], the GTM defines explicit probability density of data and aims to find an optimized model by using the Expectation-Maximization (EM) algorithm.

Although the EM algorithm [3] has been widely used to solve optimization problems in many machine learning algorithms, such as the K-Means for clustering, the EM has a severe limitation, known as the initial value problem, in which solutions can vary depending on the initial parameter setting. To overcome such a problem, we have applied the Deterministic Annealing (DA) algorithm to GTM to find more robust answers against random initial values.

The core of DA algorithm is to find an optimal solution in a deterministic way, which contrast to a stochastic way in the simulated annealing [8], by controlling the level of randomness. This process, adapted from physical annealing process, is known as a cooling schedule in that an optimal solution is gradually revealed by lowering randomness. At each level of randomness, the DA algorithm chooses an optimal solution by using the principle of maximum entropy [6, 5, 7], a rational approaches to choose the most unbiased and non-committal answers for given conditions.

The DA algorithm [10] has been successfully applied to solve many optimization problems in various machine learning algorithms and applied in many problems, such as clustering [4, 10] and visualization [9]. Ueda and Nakano has developed a general solution of using DA to solve the EM algorithms [11]. However, not many researches have been conducted to research details of processing the DA algorithm. In this paper, we will tackle down more practical aspects in using DA with GTM.

The main contributions of our paper are as follow:

- Developing the DA-GTM algorithm which uses the DA algorithm to solve GTM problem. Our DA-GTM can give more robust answers than the original GTM which uses the EM, not suffering from the random initial value problem.

- Developing an adaptive cooling schedule scheme, in which the DA-GTM algorithm can generate more reliable solutions than other conventional fixed cooling schemes.

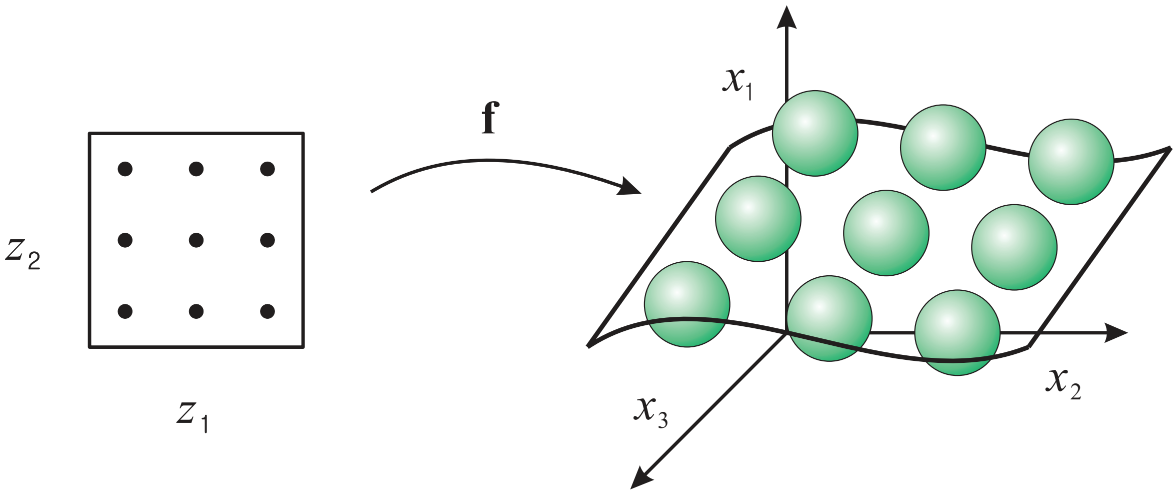


Figure 1: Non-linear embedding by GTM

- Developing an equation to compute the first phase transition temperature, which can be used to set the initial temperature of the DA-GTM algorithm. With this equation, users can choose the starting temperature directly.

2 GTM

We start by reviewing the original GTM algorithm [1]. The GTM algorithm is to find a non-linear manifold embedding of K latent variables $\boldsymbol{z}_k \in \mathbb{R}^L (k = 1, \cdots, K)$ in the latent space, which can optimally represent the given N data points $\boldsymbol{x}_n \in \mathbb{R}^D (n = 1, \cdots, N)$ in the data space (usually $L \ll D$) (Figure 1). This is achieved by two steps: First, mapping the latent variables to the data space with respect to the non-linear mapping $f : \mathbb{R}^L \mapsto \mathbb{R}^D$ such as $\boldsymbol{y}_k = f(\boldsymbol{z}_k; \boldsymbol{W})$ for a parameter matrix $\boldsymbol{W}$ with $\boldsymbol{z}_k \in \mathbb{R}^L$ and $\boldsymbol{y}_k \in \mathbb{R}^D$ (we will discuss details of this function later). Secondly, estimating probability density of data points $\boldsymbol{x}_n$ by using the Gaussian noise model in which the probability density of data point $\boldsymbol{x}_n$ is defined as an isotropic Gaussian centered on $\boldsymbol{y}_k$ having variance σ^2. I.e., the probability density $p(\boldsymbol{x}_n|\boldsymbol{y}_k)$ has the following normal distribution.

$$\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{y}_k, \sigma) \quad = \quad \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right). \tag{1}$$

The mapping $f(\boldsymbol{z}_k; \boldsymbol{W})$ can be any parametric, non-linear model. In the GTM [1], for example, $\boldsymbol{y}_k = f(\boldsymbol{z}_k; \boldsymbol{W})$ has been defined as a generalized linear regression model, where $\boldsymbol{y}_k$ is a linear combination of a set of fixed M basis functions such as,

$$\boldsymbol{y}_k = f(\boldsymbol{z}_k; \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{z}_k), \tag{2}$$

where $\boldsymbol{\phi}(\boldsymbol{z}_k) = (\phi_1(\boldsymbol{z}_k), \ldots, \phi_M(\boldsymbol{z}_k))$ is a column vector with M basis functions $\phi_m(\boldsymbol{z}_k) \in \mathbb{R}(m = 1, \ldots, M)$ and $\boldsymbol{W}$ is a $D \times M$ matrix containing weight parameters.

With K latent points $\boldsymbol{z}_k(k = 1, \ldots, K)$, the marginal probability of the data $\boldsymbol{x}_n$ can be written by:

$$\begin{aligned} p(\boldsymbol{x}_n|\boldsymbol{W}, \sigma) &= \sum_{k=1}^{K} p(\boldsymbol{x}_n|\boldsymbol{z}_k, \boldsymbol{W}, \sigma) \; p(\boldsymbol{z}_k) && (3) \\ &= \frac{1}{K} \sum_{k=1}^{K} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right) && (4) \end{aligned}$$

with the assumption of uniform marginal probability $p(\boldsymbol{z}_k) = 1/K$.

For given N data points, $\boldsymbol{x}_n \in \mathbb{R}^D$ for $n = 1, \ldots, N$, the GTM is to find an optimal parameter set $\{\boldsymbol{W}, \sigma\}$ which makes the following negative log-likelihood minimum:

$$\begin{aligned} l(\boldsymbol{W}, \sigma) &= \arg\min_{\boldsymbol{W},\sigma} -\log \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{W}, \sigma) && (5) \\ &= \arg\min_{\boldsymbol{W},\sigma} -\sum_{x=1}^{N} \log \frac{1}{K} \sum_{k=1}^{K} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right) && (6) \end{aligned}$$

Since the problem is intractable, the GTM uses the EM method as follows: starting with randomly initialized $\boldsymbol{W}$ matrix and iterating the following two equations:

$$\begin{aligned} \boldsymbol{\Phi}^t \boldsymbol{G}_{old} \boldsymbol{\Phi} \boldsymbol{W}^t_{new} &= \boldsymbol{\Phi}^t \boldsymbol{R}_{old} \boldsymbol{X} && (7) \\ \sigma^2_{new} &= \frac{1}{ND} \sum_{x=1}^{N} \sum_{k=1}^{K} r_{kn} \|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2 && (8) \end{aligned}$$

where

- $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ is the data matrix of $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^t$ where $\boldsymbol{x}_n \in \mathbb{R}^D$ for $n = 1, \ldots, N$.
- $\boldsymbol{R}$ is a matrix of $\mathbb{R}^{K \times N}$ and its element r_{kn} ($k = 1, \ldots, K$ and $n = 1, \ldots, N$), known as responsibility or posterior probability, defined by

$$r_{kn} = p(\boldsymbol{z}_k|\boldsymbol{x}_n, \boldsymbol{W}, \sigma) = \frac{p(\boldsymbol{x}_n|\boldsymbol{z}_k, \boldsymbol{W}, \sigma)}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|\boldsymbol{z}_{k'}, \boldsymbol{W}, \sigma)} \qquad (9)$$

- $\boldsymbol{G}$ is a $K \times K$ diagonal matrix with its elements $g_{kk} = \sum_{n=1}^{N} r_{kn}$
- $\boldsymbol{\Phi}$ is a $M \times K$ matrix, of which k-th column is $\boldsymbol{\phi}(\boldsymbol{z}_k)$

The solution we want to find will be converged through the iteration process. However, as observed in the K-means method, the EM in the GTM also suffers from the random initial values in which the solutions can vary depending on the initial parameters.

3 Deterministic Annealing GTM (DA-GTM)

Instead of using the EM, we can use the DA to find an optimal solution to the GTM. With the DA, we can have more robust solutions against the random initial value problem.

The core of the DA is to define an objective function, known as *free energy*, in terms of an expected cost of configurations and its entropy and to trace the global solution which minimizes the free energy function.

The problem remains for us is how to define the right free energy function to the GTM. We can do this in two different ways but both give the same result: i) following the method used by Rose [10] or ii) simply using the equations by Ueda and Nakano [11]. We will introduce both methods in the following.

3.1 Rose's method

Defining the free energy is crucial in the DA. This problem can be solved by the method used by Rose in [10]. The sketch of this procedure is as follow: First, we need to define a function D_{nk}, which represents a cost function for association between a data variable $\boldsymbol{x}_n$ and the latent variable $\boldsymbol{y}_k$, and define the free energy

$$F = \langle D_{nk} \rangle - TH, \tag{10}$$

where $\langle D_{nk} \rangle$ is the expected cost of D_{nk} for all n, k and H is the entropy for the given parameter T, also known as temperature. Secondly, find an optimal posterior distribution which minimize the free energy F.

Let define a cost for the association of two variables $\boldsymbol{x}_n$ and $\boldsymbol{y}_k$ as a function of joint probability $p(\boldsymbol{x}_n, \boldsymbol{y}_k)$. By using the GTM's Gaussian model $p(\boldsymbol{x}_n|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{y}_k, \sigma)$, the cost function D_{nk} can be defined by

$$\begin{aligned} D_{nk} &= -\log p(\boldsymbol{x}_n, \boldsymbol{y}_k) && (11)\\ &= -\log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{y}_k, \sigma) p(\boldsymbol{y}_k). && (12) \end{aligned}$$

The idea behind the cost function is that the association cost of two variables will be low when the probability $p(\boldsymbol{x}_n, \boldsymbol{y}_k)$ is high and so the configuration is easy to observe. Otherwise, the cost will increase.

With this definition, we can compute the expected cost $\langle D_{nk} \rangle$ by using the posterior probability, also known as responsibility, such that

$$\langle D_{nk} \rangle = \sum_n^N \sum_k^K p(\boldsymbol{x}_n, \boldsymbol{y}_k) d(\boldsymbol{x}_n, \boldsymbol{y}_k) \tag{13}$$

$$= \sum_n^N p(\boldsymbol{x}_n) \sum_k^K p(\boldsymbol{y}_k | \boldsymbol{x}_n) d(\boldsymbol{x}_n, \boldsymbol{y}_k). \tag{14}$$

Note that $\sum_k^K p(\boldsymbol{y}_k|\boldsymbol{x}_n) = 1$.

We can also define the entropy as $H(X, Y) = H(X) + H(Y|X)$ where the conditional entropy $H(Y|X)$ can be defined by

$$H(X|Y) = -\sum_n^N \sum_k^K p(\boldsymbol{x}_n, \boldsymbol{y}_k) \log p(\boldsymbol{y}_k | \boldsymbol{x}_n) \tag{15}$$

$$= -\sum_n^N p(\boldsymbol{x}_n) \sum_k^K p(\boldsymbol{y}_k | \boldsymbol{x}_n) \log p(\boldsymbol{y}_k | \boldsymbol{x}_n) \tag{16}$$

Now, we can define the free energy as follows:

$$F = \langle D_{nk} \rangle - TH \tag{17}$$

$$= \sum_n^N p(\boldsymbol{x}_n) \sum_k^K p(\boldsymbol{y}_k | \boldsymbol{x}_n) d(\boldsymbol{x}_n, \boldsymbol{y}_k) + \sum_n^N p(\boldsymbol{x}_n) \sum_k^K p(\boldsymbol{y}_k | \boldsymbol{x}_n) \log p(\boldsymbol{y}_k | \boldsymbol{x}_n) + \sum_n^N \sum_k^K p(\boldsymbol{x}_n \tag{18}$$

However, we don't know yet what kind of posterior probability $p(\boldsymbol{y}_k|\boldsymbol{x}_n)$ will minimize the free energy (Eq. 17). We can solve this optimization problem by using the following Lagrangian equation with the constraint $\sum_k^K p(\boldsymbol{y}_k|\boldsymbol{x}_n) = 1$:

$$F^* = \langle D_{nk} \rangle - TH + \lambda_x \left(\sum_k^K p(\boldsymbol{y}_k | \boldsymbol{x}_n) - 1 \right) \tag{19}$$

for λ_x is a Lagrange multiplier.

When $\partial F / \partial p(\boldsymbol{y}_k|\boldsymbol{x}_n) = 0$, we get the optimal posterior distribution as

$$p(\boldsymbol{y}_k | \boldsymbol{x}_n) = \frac{\mathcal{N}(\boldsymbol{x}_n | \boldsymbol{y}_k, \sigma)^{\frac{1}{T}}}{Z_x}, \tag{20}$$

where the normalizer $Z_x = \sum_{\boldsymbol{y}_k} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{y}_k, \sigma)^{\frac{1}{T}} p(\boldsymbol{y}_k)^{\frac{1}{T}}$, which is called as the partition function.

Note that the optimal distribution we get (Eq. 20) is also known as the Gibbs distribution [10].

3.2 Solution with Ueda and Nakano's equations

We can derive the free energy and the optimal distribution directly, if we use the equations in [11] by Ueda and Nakano.

With the same cost function defined in Eq. (11), we can compute a Gibbs distribution by

$$\begin{aligned} p^{Gb}(D_{nk}) &= \frac{\exp\left(-\frac{1}{T}D_{nk}\right)}{Z_x} & (21)\\ &= \exp\left\{\frac{1}{T}\log\left(\frac{1}{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{y}_k,\sigma)\right)\right\}/Z_x & (22)\\ &= \left(\frac{1}{K(2\pi\sigma^2)^{D/2}}\right)^{\frac{1}{T}}\exp\left\{-\frac{1}{2\sigma^2 T}\|\boldsymbol{x}_n-\boldsymbol{y}_k\|^2\right\}/Z_x & (23)\end{aligned}$$

where

$$\begin{aligned} Z_x &= \sum_{k'=1}^{K}\exp\left(-\frac{1}{T}D_{k'n}\right) & (24)\\ &= \sum_{k'=1}^{K}\left(\frac{1}{K(2\pi\sigma^2)^{D/2}}\right)^{\frac{1}{T}}\exp\left\{-\frac{1}{2\sigma^2 T}\|\boldsymbol{x}_n-\boldsymbol{y}_{k'}\|^2\right\} & (25)\end{aligned}$$

which is known as partition function and T is known as temperature.

Now we can define the free energy as follows:

$$\begin{aligned} F(\boldsymbol{W},\sigma,T) &= -T\sum_{n}^{N}\log Z_x & (26)\\ &= -T\sum_{n}^{N}\log\sum_{k}^{K}\left(\frac{1}{K(2\pi\sigma^2)^{D/2}}\right)^{\frac{1}{T}}\exp\left\{-\frac{1}{2\sigma^2 T}\|\boldsymbol{y}_k-\boldsymbol{x}_n\|^2\right\} & (27)\\ &= -T\sum_{n}^{N}\log\sum_{k}^{K}\left(\frac{1}{K}\right)^{\frac{1}{T}}p(\boldsymbol{x}_n|\boldsymbol{z}_k,\boldsymbol{W},\sigma)^{\frac{1}{T}} & (28)\end{aligned}$$

which we want to minimize as $T \rightarrow 0$.

Note that the GTM's log-likelihood function (Eq 5), which is a target to minimize in GTM, differs only the use of temperature T with our free energy function $F(\boldsymbol{W},\sigma,T)$. Especially, at $T = 1$, $l(\boldsymbol{W},\sigma) = F(\boldsymbol{W},\sigma,T)$ and so GTM's target function can be considered as a special case of the DA-GTM's.

3.3 Deterministic Annealing optimization

Now we want to minimize $F(\boldsymbol{W}, \sigma, T)$. Let $p_{kn} = p(\boldsymbol{x}_n | \boldsymbol{z}_k, \boldsymbol{W}, \sigma)$ and then,

$$\begin{aligned}
\frac{\partial F}{\partial \boldsymbol{w}_i} &= -T \left(\frac{1}{K}\right)^{\frac{1}{T}} \sum_n^N \frac{\sum_k^K \frac{1}{T}(p_{kn})^{\frac{1}{T}} \frac{1}{\sigma^2}(t_{ni} - \boldsymbol{w}_i^t \boldsymbol{y}_k)\boldsymbol{y}_k}{\sum_k^K (p_{kn})^{\frac{1}{T}}} && (29)\\
&= -T \left(\frac{1}{K}\right)^{\frac{1}{T}} \sum_n^N \sum_k^K \frac{1}{T}(r_{kn})^{\frac{1}{T}} \frac{1}{\sigma^2}(t_{ni} - \boldsymbol{w}_i^t \boldsymbol{y}_k)\boldsymbol{y}_k && (30)
\end{aligned}$$

and

$$\frac{\partial F}{\partial \sigma} = T \left(\frac{1}{K}\right)^{\frac{1}{T}} \sum_n^N \sum_k^K \frac{1}{T}(r_{kn})^{\frac{1}{T}} \left(\frac{D\sigma^2}{2} - \frac{1}{2}\|\boldsymbol{y}_k - \boldsymbol{x}_n\|^2\right) \qquad (31)$$

where $\boldsymbol{w}_i$ is the i-th column vector of $\boldsymbol{W}$. Both derivatives should be zero at an optimal point.

Parameters $\boldsymbol{W}$ and σ can be computed by EM similarly with GTM but with using additional temperature parameter T as follows:

$$\begin{aligned}
\boldsymbol{\Phi}^t \boldsymbol{G'}_{old} \boldsymbol{\Phi} \boldsymbol{W}^t_{new} &= \boldsymbol{\Phi}^t (\boldsymbol{R}_{old})^{\frac{1}{T}} \boldsymbol{X} && (32)\\
\sigma^2_{new} &= \frac{1}{D} \frac{\sum_n^N \sum_k^K r_{kn}^b \|\boldsymbol{y}_k - \boldsymbol{x}_n\|^2}{\sum_n^N \sum_k^K (r_{kn})^{\frac{1}{T}}} && (33)
\end{aligned}$$

where $\boldsymbol{G'}$ is a $K \times K$ diagonal matrix with elements $g'_{kk} = \sum_n^N (r_{kn})^{\frac{1}{T}}$

4 Phase Transitions of DA-GTM

As a characteristic behavior of the DA algorithm, explained by Rose in [10], the DA algorithm undergoes phase transitions as lowering the temperatures. At some temperature in the DA, we can not obtain all solutions but, instead, we can only obtain effective number of solutions. All solutions will gradually pop out while the annealing process proceeds as with lowering the temperature.

In the DA-GTM, we can observe the same behavior. As an extreme example, at very high temperature, the DA-GTM gives only one effective latent point in which all $\boldsymbol{y}_k$'s are converged into the same point which is the center of data points, such that $\boldsymbol{y}_k = \sum_{n=1}^N \boldsymbol{x}_n / N$. As lowering the temperature under a certain point, $\boldsymbol{y}_k$'s settled in the same point start to "explode". We call this temperature as the first critical temperature, denoted by T_c. As we further lowering temperature, we can observe subsequent phase transitions and so existence of multiple critical temperatures. Computing the first phase transition is an important task since we should begin our annealing process with the starting temperature bigger than T_c.

In the DA, we can define such phase transitions as a moment of loosing stability of the objective function, the free energy F, and turning to be unstable. Mathematically, that moment corresponds to the point in which the Hessian of the object function looses its positive definiteness.

In the DA-GTM, we can have the following Hessian matrix as a block matrix:

$$H = \begin{bmatrix} H_{11} & \cdots & H_{1K} \\ \vdots & & \vdots \\ H_{K1} & \cdots & H_{KK} \end{bmatrix}, \tag{34}$$

where an element H_{ij} is a sub matrix representing a second derivative of the free energy F as in Eq. (26), defined by $\frac{\partial^2 F}{\partial \boldsymbol{y}_i \partial \boldsymbol{y}_j}$ for $i, j = 1, \ldots, K$. More specifically, we can derive H_{ij} from the definition of the free energy in Eq. (26) as follows:

$$H_{ii} = \frac{\partial^2 F}{\partial \boldsymbol{y}_i^2} \tag{35}$$

$$= -\sum_n^N \left\{ \frac{\beta^2}{T} r_{in}(1 - r_{in})(\boldsymbol{x}_n - \boldsymbol{y}_i)^t (\boldsymbol{x}_n - \boldsymbol{y}_i) - \beta r_{in} \boldsymbol{I}_D \right\} (\text{if } i = j), \text{or} \tag{36}$$

$$H_{ij} = \frac{\partial^2 F}{\partial \boldsymbol{y}_i \partial \boldsymbol{y}_j} \tag{37}$$

$$= \sum_n^N \left\{ \frac{\beta^2}{T} r_{in} r_{jn} (\boldsymbol{x}_n - \boldsymbol{y}_i)^t (\boldsymbol{x}_n - \boldsymbol{y}_j) \right\} (i \neq j), \tag{38}$$

where $i, j = 1, \ldots, K$, $\beta = 1/\sigma^2$, and $\boldsymbol{I}_D$ is an identity matrix of size D. Note that H_{ij} is a $D \times D$ matrix and thus, $H \in \mathbb{R}^{KD \times KD}$.

To compute the first phase transition, let us start with a very simple system in which we have only two latent point $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Assuming that the system hasn't undergone the first phase transition and the current temperature is high enough so that two point $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are settled in the center of the data point, denoted by $\boldsymbol{y}_0$, such as $\boldsymbol{y}_0 = \boldsymbol{y}_1 = \boldsymbol{y}_2 = \sum_n^N \boldsymbol{x}_n / N$ and thus all the responsibilities are same, such as $r_{1n} = r_{2n} = 1/2$ for all $n = 1, \ldots, N$.

In this simple system, the second derivatives of the free energy F can be defined by the following:

$$H_{11} = H_{22} = -\frac{\beta^2 N}{4T} \left(\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} - \frac{2T}{\beta} \boldsymbol{I}_D \right) \tag{39}$$

$$H_{12} = H_{21} = \frac{\beta^2 N}{4T} \left(\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \right) \tag{40}$$

where $\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0}$ represents a covariance matrix of centered data set such that,

$$\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} = \frac{1}{N} \sum_{n=1}^N (\boldsymbol{x}_n - \boldsymbol{y}_0)^t (\boldsymbol{x}_n - \boldsymbol{y}_0) \tag{41}$$

Then, the Hessian matrix in this system can be defined by

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{11} \end{bmatrix}, \tag{42}$$

and its determinant can be computed as follows:

$$\det(H) = \det\left(-\frac{\beta^2 N}{4T}\left\{\begin{bmatrix} \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \\ -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \end{bmatrix} - \frac{2T}{\beta}\boldsymbol{I}_D\right\}\right) \tag{43}$$

$$= \left(\frac{-\beta^2 N}{4T}\right)^{2D} \det\left(\begin{bmatrix} \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \\ -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \end{bmatrix} - \frac{2T}{\beta}\boldsymbol{I}_D\right) \tag{44}$$

The first phase transition occurs when the system is getting unstable so that the above Hessian matrix is loosing its positive definiteness. I.e., the first phase transition is the moment when the Hessian matrix becomes singular and so its determinant equals 0(zero), such that $\det(H) = 0$ at $T = T_c$, which holds the following:

$$\text{eig}\left(\begin{bmatrix} \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \\ -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \end{bmatrix}\right) = \frac{2T_c}{\beta} \tag{45}$$

where $\text{eig}(\boldsymbol{A})$ is an eigenvalue of $\boldsymbol{A}$.

We can further simplify the above equation by using the Kronecker product:

$$\text{eig}\left(\begin{bmatrix} \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \\ -\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} & \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0} \end{bmatrix}\right) = \text{eig}\left(\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \otimes \boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0}\right) \tag{46}$$

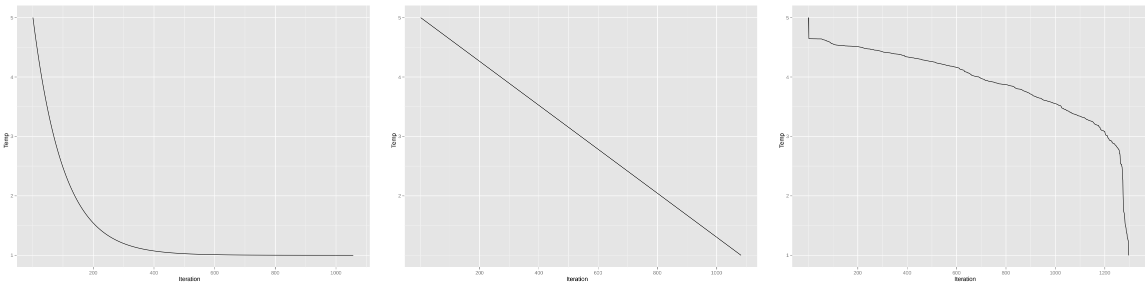
$$= \text{eig}\left(\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\right) \otimes \text{eig}\left(\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0}\right) \tag{47}$$

Since the first critical temperature is the most largest one, we can only use the maximum eigenvalue. Thus, the first critical temperature can be obtained by the following:

$$T_c = \beta\lambda_{\max} \tag{48}$$

where $\lambda_{\max}$ is the largest value of $\text{eig}(\boldsymbol{S}_{\boldsymbol{x}|\boldsymbol{y}_0})$ and $\beta = 1/\sigma^2$ can be computed from Eq. (8), such as

$$\beta = \frac{ND}{\sum_n^N (\boldsymbol{x}_n - \boldsymbol{y}_0)^2} \tag{49}$$



(a) exponential (b) linear (c) adaptive

Figure 2: Cooling Schedules

5 Adaptive cooling schedule

The DA has been applied in many applications and proved its success to find global optimal solutions by avoiding local minimum. However, up to our knowledge, no literature has been found to research on the cooling schedule. Commonly used cooling schedule is exponential, such as $T = \alpha T$, or linear, such as $T = T - \delta$. Those scheduling schemes are fixed in that temperatures are pre-defined and the constant δ or α will not be changed during the process, regardless of the complexity of a given problem.

However, as we discussed previously, the DA algorithm undergoes the phase transitions in which the solution space can change dramatically. One may try to use very small δ near 0 or $alpha$ near 1 to avoid such drastic changes but the procedure can go too long to be used in practice.

To overcome such problem, we propose an adaptive cooling schedule in which next cooling temperatures are determined dynamically in the annealing process. More specifically, at every iteration of DA algorithm, we predict the next phase transition temperature and move to the point as quickly as possible. Figure 2 shows an example, comparing fixed cooling schedules ((a) and (b)) versus an adaptive cooling schedule.

Computing the next critical temperature T is very similar with the way to compute the first critical temperature in the previous section, except that now we need to consider K points, usually $K > 2$.

With $K(K > 2)$ points, the size of Hessian matrix is KD-by-KD, which is too big to be used in practice. Instead, we can consider much smaller Hessian matrix for each k-th point $\boldsymbol{y}_k$. The sketch of the algorithm to find the next critical temperature as follows: i) For each point $\boldsymbol{y}_k (k = 1, \ldots, K)$, we add an imaginary point $y_{k'}$ which is an exact replica of $\boldsymbol{y}_k$ such that $\boldsymbol{y}_{k'} = \boldsymbol{y}_k$ and thus it shares too the responsibility $r_{kn} = r_{k'n} = 1/2 r_{kn}$. ii) Then, find the possible critical temperature T_c which is lower than the current temperature T and also makes the following Hessian matrix to be singular, such that

$$\det\left(\left[\begin{array}{cc} H_{kk} & H_{kk'} \\ H_{kk'} & H_{kk} \end{array}\right]\right) = 0 \tag{50}$$

where

$$\begin{aligned} H_{kk} &= -\sum_{n}^{N}\left\{\frac{\beta^2}{T}r_{kn}(1-r_{kn})(\boldsymbol{x}_n-\boldsymbol{y}_k)^t(\boldsymbol{x}_n-\boldsymbol{y}_k)-\beta r_{kn}\boldsymbol{I}_D\right\} & (51)\\ H_{kk'} &= \sum_{n}^{N}\left\{\frac{\beta^2}{T}r_{kn}r_{k'n}(\boldsymbol{x}_n-\boldsymbol{y}_k)^t(\boldsymbol{x}_n-\boldsymbol{y}_{k'})\right\}, & (52)\end{aligned}$$

where again $\boldsymbol{I}_D$ is an identity matrix of size D. iii) Finally, among all T_c for $k = 1, \ldots, K$, choose the largest T_c, which should satisfy $T_c < T$.

To simplify the above equations, define the following:

$$\begin{aligned} \boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k} &= \sum_{n=1}^{N} r_{kn}(\boldsymbol{x}_n-\boldsymbol{y}_k)^t(\boldsymbol{x}_n-\boldsymbol{y}_k) & (53)\\ \boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} &= \sum_{n=1}^{N} (r_{kn})^2(\boldsymbol{x}_n-\boldsymbol{y}_k)^t(\boldsymbol{x}_n-\boldsymbol{y}_k) & (54)\\ g_k &= \sum_{n=1}^{N} r_{kn} & (55)\end{aligned}$$

Then, we can redefine the second derivatives

$$\begin{aligned} H_{kk} &= \frac{-\beta^2}{T}\left(\frac{\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k}}{2}-\frac{\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k}}{4}\right)+\frac{\beta g_k}{2}\boldsymbol{I}_D & (56)\\ &= \frac{-\beta^2}{4T}\left(2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k}-\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k}-\frac{2Tg_k}{\beta}\boldsymbol{I}_D\right) & (57)\\ H_{kk'} &= \frac{\beta^2}{4T}\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & (58)\end{aligned}$$

The determinant of the Hessian as in Eq. (50) can be computed by

$$\left(\frac{-\beta^2}{4T}\right)^{2D}\det\left(\left[\begin{array}{cc} 2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k}-\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & -\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} \\ -\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & 2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k}-\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} \end{array}\right]-\frac{2Tg_k}{\beta}\boldsymbol{I}_{2D}\right), \tag{59}$$

where $\boldsymbol{I}_{2D}$ is an identity matrix of size $2D$.

As before, when $T = T_c$, the above equation will equal 0(zero) and so the following equation

```
Next-Critical-Temperature
for $k \leftarrow 1$ to $K$
        do
            ▷ Define a duplicated point $\boldsymbol{y}_{k'}$
            $\boldsymbol{y}_{k'} \leftarrow \boldsymbol{y}_k$
            Compute $T_c$ by using Eq (61)
$T \leftarrow \max(T_c)$
```

Figure 3: Pseudo code for find the next critical temperature

holds:

$$\mathrm{eig}\left(\begin{bmatrix} 2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k} - \boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & -\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} \\ -\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & 2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k} - \boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} \end{bmatrix}\right) = \frac{2T_c g_k}{\beta} \tag{60}$$

Thus, the next critical temperature T_c at $\boldsymbol{y}_k$ is

$$T_c = \frac{\beta}{2g_k}\lambda_{\text{next}} \tag{61}$$

where λ_{next} is the largest eigenvalue among the eigenvalues of the matrix

$$\begin{bmatrix} 2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k} - \boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & -\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} \\ -\boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} & 2\boldsymbol{U}_{\boldsymbol{x}|\boldsymbol{y}_k} - \boldsymbol{V}_{\boldsymbol{x}|\boldsymbol{y}_k} \end{bmatrix} \tag{62}$$

and smaller than $2g_k T/\beta$ so that $T_c < T$.

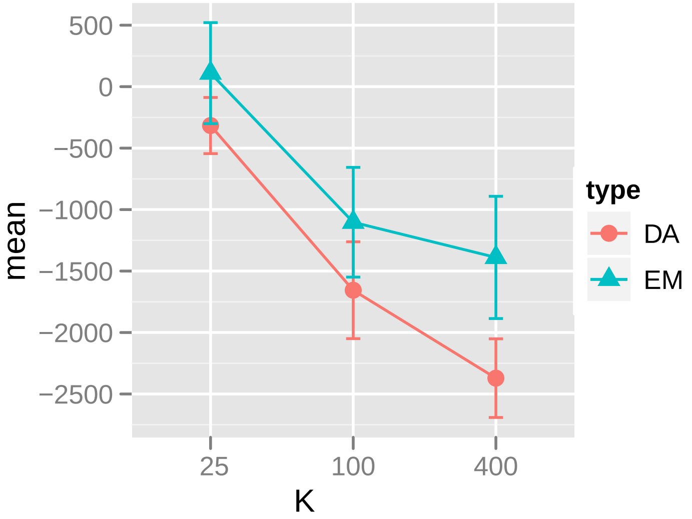
The overall pseudo algorithm is shown in Figure 3.

6 Experiment Results

To compare performances of DA-GTM with the original GTM, EM-GTM, we performed a set of experiments by using the same data set used in original GTM paper [2]. The data set is known as oil flow data which were synthetically generated by drawing with equal probability from the 3 configurations and consists of 1,000 points in 12-dimensional.

6.1 Robustness

DA-GTM is robust against random initialization from which original GTM suffers. To show DA-GTM's robustness, we run randomly initialized 100 executions of DA-GTM and EM-GTM with the same data set and measured means and standard deviations of both. As the result shows in



		EM		DA	
Model Size	Grid Size	Mean	SD	Mean	SD
9	25	110.035	410.802	-316.276	228.324
9	100	-1102.940	446.318	-1655.820	393.988
9	400	-1389.106	497.113	-2371.191	319.951

Figure 4: Comparison of robustness against random initialization. For 100 randomly initialized runs, DA-GTM produced more optimized answers (lower mean llh) with smaller deviations.

Figure 4, DA-GTM produced more optimized answers (lower mean log likelihood) with smaller deviation.

References

[1] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: A principled alternative to the self-organizing map. Advances in neural information processing systems, pages 354–360, 1997.

[2] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: The generative topographic mapping. Neural computation, 10(1):215–234, 1998.

[3] A.P. Dempster, N.M. Laird, D.B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38, 1977.

[4] T. Hofmann and J.M. Buhmann. Pairwise data clustering by deterministic annealing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(1):1–14, 1997.

[5] E.T. Jaynes. Information theory and statistical mechanics. II. Physical review, 108(2):171–190, 1957.

[6] ET Jaynes. Information theory and statistical methods I. Physics Review, 106(1957):620–630, 1957.

[7] ET Jaynes. On the rationale of maximum-entropy methods. Proceedings of the IEEE, 70(9):939–952, 1982.

[8] S. Kirkpatric, CD Gelatt, and MP Vecchi. Optimization by simulated annealing. Science, 220(4598):671–680, 1983.

[9] H. Klock and J.M. Buhmann. Multidimensional scaling by deterministic annealing. Lecture Notes in Computer Science, 1223:245–260, 1997.

[10] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proceedings of the IEEE, 86(11):2210–2239, 1998.

[11] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. Neural Networks, 11(2):271–282, 1998.