Introduction to the DREAM Algorithm

Chuhan Shao

1 Introduction

Differential Evolution Adaptive Metropolis, DREAM, algorithm is proposed by Vrugt et al. to run a global exploration and automatically tunes the scale and orientation of the proposal distribution [1].

The *population state* $\mathbf{X}$ is represented by a set of P d-dimensional points, and one *step* generates a new population state. The samples are produced gradually during N steps, where there are P chains in a row and N steps in a column. Figure 1 shows the conceptual structure used in the DREAM algorithm.

The DREAM algorithm starts with a randomly generated population state $\mathbf{X}$. For each point $\mathbf{x}^{(p)}$, a series of methods are applied sequentially, called a *chain evolution*. Section 2 to Section 5 illustrates how DREAM algorithm proposes a candidate for a single chain. Then Section 6 shows the method to update an algorithm parameter after a chain evolution, while Section 7 introduces a method to speed up convergence of chains.

2 Generate the Pre-candidate $\mathbf{x}_d^*$

Assume that $\mathbf{x}^{(p)}$ is the p-th chain in the population state $\mathbf{X}$, Equation 1 is used to generate a d-dimensional pre-candidate $\mathbf{x}_d^*$ for $\mathbf{x}^{(p)}$.

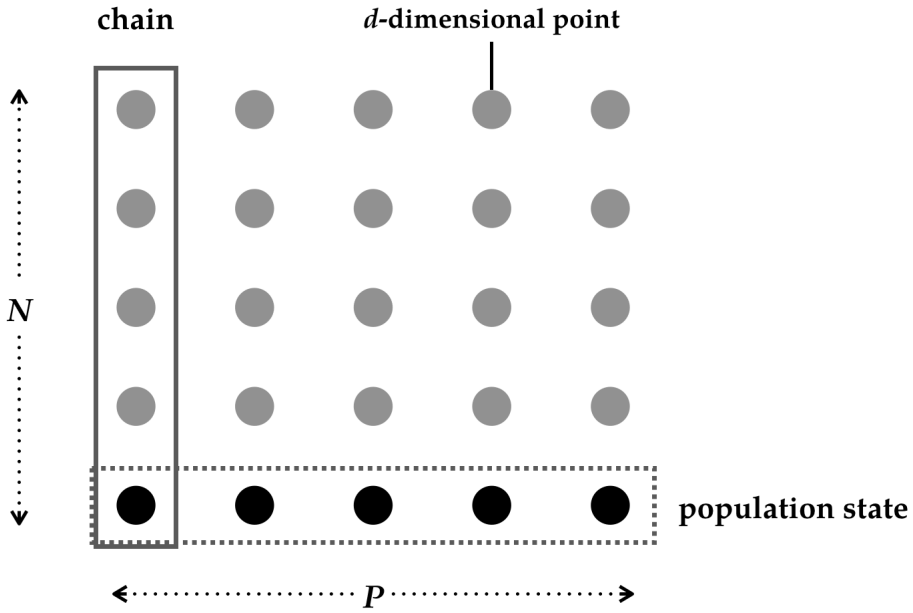

Figure 1: Conceptual structure for DREAM algorithm.

$$\mathbf{x}_d^* = \mathbf{x}^{(p)} + \sigma(1+e)I_d \sum_{j=1}^{\delta^*}(\mathbf{x}^{(a_j)} - \mathbf{x}^{(b_j)}) + \varepsilon_d \tag{1}$$

where $\delta^* \in \{1, 2, \ldots, \delta\}$ signifies the number of pairs used to make differences for generating the pre-candidate with δ, a user-defined parameter. And for $j = 1, 2, \ldots, \delta^*$, $a_j, b_j \in \{1, 2, \ldots, P\}$, while $a_j \neq b_j \neq p$. σ is the jump ratio, and $\sigma = 2.38/\sqrt{2\delta^* d}$. $e \in \mathcal{U}_d(-c, c)$ and $\varepsilon_d \in \mathcal{N}_d(0, c^*)$ with c and c^* small compared to the scale of the target distribution.

3 Subspace Sampling

Subspace sampling is a technique for randomly choosing one or many subspaces in the whole space $\mathbb{R}^d$. A is the subset of d^*-dimension drawn from the original d-dimensional parameter space, where the parameter d^* $(0 < d^* \leq d)$ denotes the number of selected dimensions, i.e., $d^* = |A|$, with $\mathbb{R}^{d^*} \subseteq \mathbb{R}^d$.

A Crossover Rate (CR) value is sampled from an *arithmetic progression* of n_{CR} different CR values, $CRs = \{\frac{1}{n_{CR}}, \frac{2}{n_{CR}}, \ldots, 1\}$. A discrete *categorical distribution* $\mathbf{p_{CR}} = (p_{CR,1}, p_{CR,2}, \ldots, p_{CR,n_{CR}})$ represents the probability of seeing the i-th element in CR. Let θ be an arbitrary element in CRs and its position in CRs is i $(1 \leq i \leq n_{CR})$, while $\sum_{i=1}^{n_{CR}} p_{CR,i} = 1$. The probability Pr that the selected CR vlaue $CR = \theta$ can be illustrated as Equation 2. $\mathbf{p_{CR}}$ is initially set as a discrete *uniform distribution* and is then updated according to the *standardized squared average jump* distance under each CR value, which is introduced in section 6.

$$Pr(CR = \theta | \mathbf{p_{CR}}) = p_i \tag{2}$$

Equation 2 can be rewritten as Equation 3, which facilitates mathematical manipulations.

$$Pr(CR | \mathbf{p_{CR}}) = \prod_{i=1}^{n_{CR}} p_{CR,i}^{[CR=\theta]}, \tag{3}$$

where $[CR = \theta]$ evaluates to 1 if $CR = \theta$, and 0, otherwise.

For an arbitrary dimension d' in the whole space d, a number μ is drawn from a uniform distribution, $\mu \sim \mathcal{U}(0, 1)$. If $\mu < \theta$, the dimension d' is chosen as a subspace and stored into a set A as shown in Equation 4. The process above executes for all the dimensions in d and a set of subspaces A is generated eventually.

$$A \leftarrow \begin{cases} A \cup \{d'\} & \text{if } \mu < \theta \\ A & \text{otherwise} \end{cases} \tag{4}$$

A formal description of the subspace sampling algorithm is listed in Algorithm 1. Note that $\mathbf{p_{CR}}$ is initialized as a discrete uniform distribution and is

then updated.

Algorithm 1 The Subspace sampling algorithm

1: Select a threshold θ in CRs under the probability $\mathbf{p_{CR}}$, $\theta \leftarrow CR$.
2: Set A as an empty set, $A \leftarrow \emptyset$.
3: **for** each d' in d **do**
4: Sample a value μ from 0 to 1, $\mu \sim \mathcal{U}(0, 1)$.
5: **if** $u < \theta$ **then**
6: Add dimension d' to the set A, $A \leftarrow A \cup \{d'\}$.
7: **Return** A.

4 Generate the Candidate $\mathbf{x}^*$

For a chain $\mathbf{x}^{(p)}$, components in the set of subspaces A are replaced by corresponding elements in the pre-candidate $\mathbf{x}_d^*$ and the rest of components remain unchanged.

A set including the chosen subspaces A is generated after subspace sampling. Two $d \times d$ diagonal matrices, S_A and $S_{\bar{A}}$, are then constructed by *one-hot encoding* [2] according to the set A and its compliment $\bar{A}$. For the i-th diagonal component in S_A, if $i \in A$, the component is 1, 0 otherwise, as shown in Equation 5.

$$S_{A,i,i} = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{if } i \notin A \end{cases} \tag{5}$$

On the contrary, for the i-th diagonal component in $s_{\bar{A}}$, if $i \notin A$, the component is 1, 0 otherwise, as shown in Equation 6.

$$S_{\bar{A},i,i} = \begin{cases} 1 & \text{if } i \notin A \\ 0 & \text{if } i \in A \end{cases} \tag{6}$$

Then, a candidate $\mathbf{x}^*$ is generated by taking the components of $\mathbf{x}_d^*$ in the subspace A and the components of $\mathbf{x}^{(p)}$ in the subspace $\bar{A}$, as shown in Equation 7.

$$\mathbf{x}^* = [S_{\bar{A}} \;\; S_A] \begin{bmatrix} \mathbf{x}^{(p)} \\ \mathbf{x}_d^* \end{bmatrix} \tag{7}$$

5 Update the Point

The acceptance ratio α for the DREAM algorithm is calculated in Equation 8.

$$\alpha(\mathbf{x}^*, \mathbf{x}^{(p)}) = \min\left(1, \frac{\pi(\mathbf{x}^*)}{\pi(\mathbf{x}^{(p)})}\right) \tag{8}$$

A uniformly random number u is then uniformly generated in the interval $[0, 1]$, where $u \sim \mathcal{U}(0, 1)$. The proposal $\mathbf{x}^*$ is then accepted or rejected by Equation 9 and the next generation chain $\mathbf{x}_{n+1}^p$ is generated.

$$\mathbf{x}_{n+1}^{(p)} = \begin{cases} \mathbf{x}^* & \text{if } u \leq \alpha \\ \mathbf{x}^{(p)} & \text{if } u > \alpha \end{cases} \tag{9}$$

6 Update Probabilities of CR

Assume that k is the denotation of dimensions, ranging from 1 to d. The jump distance from the last step $n-1$ to the current step n is illustrated as $\mathbf{x}_{n,k}^{(p)} - \mathbf{x}_{n-1,k}^{(p)}$ in terms of dimension k. After a chain is updated, the squared normalized distance is calculated as Equation 10. An n_{CR}-vector $\Delta = \{\Delta_1, \Delta_2, \ldots, \Delta_{n_{CR}}\}$

stores the squared normalized jump distance for each CR value. f_{CR} is an n_{CR}-vector to record the frequency of each CR value being selected.

$$\Delta_i \leftarrow \Delta_i + \sum_{k=1}^{d} \left(\frac{\mathbf{x}_{n,k}^{(p)} - \mathbf{x}_{n-1,k}^{(p)}}{\sigma_k} \right)^2 , \tag{10}$$

where k is the index of dimensions from 1 to d, and i is the index of CR value selected in subspaces A. Besides, σ_k denotes the current standard deviation of all the history samples in dimension k.

The normalized jump distance is a scaled length of difference $\mathbf{x}_{n,k}^{(p)} - \mathbf{x}_{n-1,k}^{(p)}$ compared to the distribution of the population state. By normalization, the jump distance is zoomed into a length which fits the scale of the population state. Squaring the normalized jump distance is to make all the values of distances positive.

The probabilities of CR $\mathbf{p_{CR}}$ is then updated by Equation 11 and 12 after a chain evolution.

$$\mathbf{p_{CR}} \leftarrow \frac{\Delta}{f_{CR}} \tag{11}$$

$$\mathbf{p_{CR}} \leftarrow \frac{\mathbf{p_{CR}}}{\sum_{i=1}^{n_{CR}} \mathbf{p_{CR}}} \tag{12}$$

Equation 11 calculates the average squared normalized jumping distance for each CR value. In other words, it is the distance that the point jumps averagely under a certain CR value. Large jump distances tend to be selected, so the probability of selecting the corresponding CR values should be enlarged. However, the sum of $\mathbf{p_{CR}}$ might not be one after updating, and Equation 12 is thus used for *standardization*.

Algorithm 2 Probabilities of CR update algorithm

1: **for** each p in P **do**

2: Select the threshold θ and update the point $\mathbf{x}_n^{(p)}$.

3: Record the fixed jump distance using Equation 10.

4: Mark the selected index in f_{CR}, $f_{CR,i} \leftarrow f_{CR,i} + 1$.

5: Reassess $\mathbf{p_{CR}}$ regarding to Equation 11.

6: Standardlize $\mathbf{p_{CR}}$ by Equation 12.

7: **Return** $\mathbf{p_{CR}}$.

7 Correction for Outlier Chains

An outlier is an observation point that is distant from other observations in statistics. The outlier chains are the points which deviate the centre of the distribution significantly. They lead the result to a situation where each chain is out of target distribution and slows down the speed of convergence.

Inter-Quartile-Range(IQR) is often used to find outliers. It is equal to the difference between upper and lower quartiles, $IQR = Q3 - Q1$. It can be clearly illustrated in a box plot, as shown in Figure 2[1] [3]. The outliers are defined as observations that fall below $Q1 - 1.5IQR$ or above $Q3 + 1.5IQR$.

In this thesis, Ω is *the mean of the logarithm of the posterior densities of the last* 50% *of the samples in each chain.* Ω can thus be regarded as the centre of the logarithm of the distribution for each chain. And chains with $\Omega < Q1 - 1.5IQR$ are considered as outliers[2] and are moved to the current best member of the population state, as shown in Figure 3.

[1]Minor modifications are applied to the original figure.

[2]For the part $\Omega > Q3 + 1.5IQR$, they are much closer to the centre of the target distribution, which is we expect them to be. They consequently are not regarded as outliers.

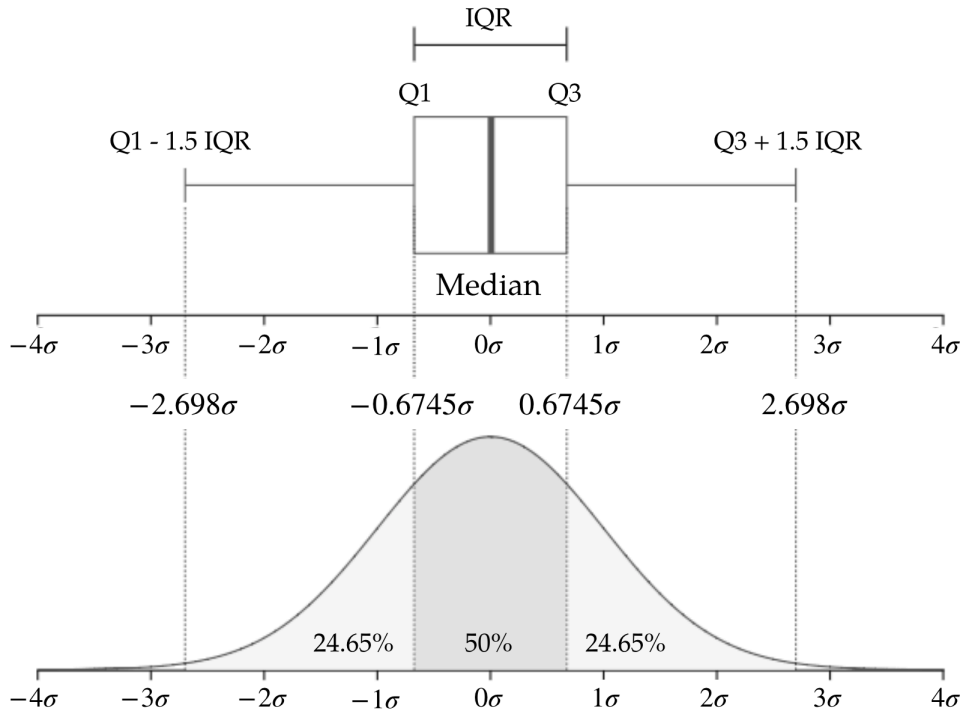

Figure 2: Boxplot and a probability density function of a Normal population

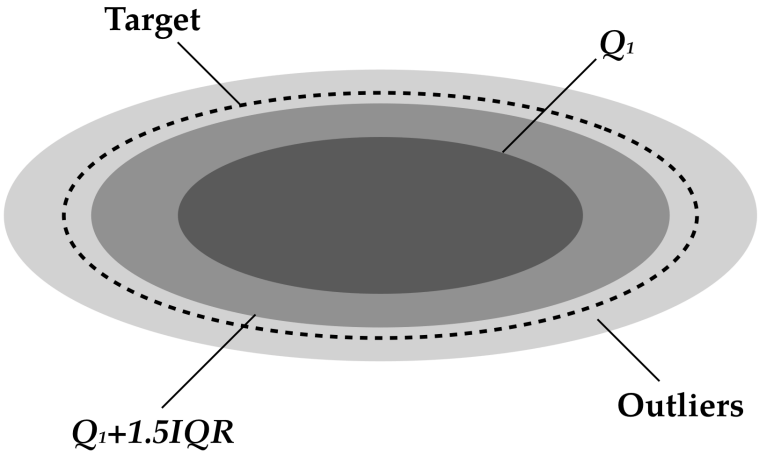

Figure 3: Outliers and the target distribution.

References

[1] Jasper A Vrugt, CJF Ter Braak, CGH Diks, Bruce A Robinson, James M Hyman, and Dave Higdon. Accelerating markov chain monte carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):273–290, 2009.

[2] Tianqi Chen. Introduction to boosted trees. *University of Washington Computer Science*, 22:115, 2014.

[3] Jhguch. Boxplot (with an interquartile range) and a probability density function (pdf) of a normal population. `https://en.wikipedia.org/wiki/Interquartile_range`. Accessed 02 August, 2019.