Weighted-kernel Deterministic Annealing algorithm to shape clustering

Mayank Baranwal

May 12, 2017

1 INTRODUCTION

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a *cluster*) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a method of *unsupervised* learning to draw inferences from datasets consisting of input data without labeled responses. Clustering is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of similarity which is defined upon metrics such as Euclidean, pairwise or probabilistic distance.

The problem is computationally difficult (**NP**-hard), and thus the common approach is to search only for approximate solutions. A particularly well known approximation method is Lloyd's algorithm,[3] often actually referred to as "k-means algorithm". It does however only find a local optimum, and is commonly run multiple times with different random initializations. On the contrary, Rose in [1] proposed an annealing-based algorithm, well described in terms of laws such as minimum free energy principle in statistical physics literature, and showed that the solutions obtained using this approach are totally independent of the choice of initial configurations. The algorithm is referred as *deterministic annealing* (DA) algorithm and is aimed to provide high-quality solutions to a clustering problem with only marginal increase in computational complexity.

The original DA algorithm by Rose aims to cluster a set of points $\mathscr{X} = \{\boldsymbol{x_i} \in \mathscr{D} : 1 \leq i \leq N\}$ in some bounded domain $\mathscr{D}$ in the Euclidean space, i.e., the points are separable in the Euclidean domain. However, a major drawback to DA algorithm is that it cannot separate clusters that are non-linearly separable in input space. On the other hand, the task of unsupervised grouping of shapes (also known as *shape clustering*) requires a non-linear separation of data points. Two recent approaches have emerged for tackling such a problem. One is kernel k-means [6], where, before clustering, points are mapped to a higher-dimensional feature space using a nonlinear function, and then kernel k-means partitions the points by linear separators in the new space. The other approach is spectral clustering algorithms [5], which use the eigenvectors of an affinity matrix to obtain a clustering of the data. A popular objective function used in spectral clustering is to minimize the normalized cut [4].

Similar to basic k-means, the kernel k-means is sensitive to initialization and a poor initialization may result in undesirable clustering performance. On the other hand, spectral clustering methods are eigenvector-based algorithms and software to compute eigenvectors of large sparse matrices (often based on the Lanczos algorithm) can have substantial computational overheads, especially when a large number of eigenvectors are to be computed. To overcome these limitations, a weighted kernel-DA approach is presented in this report. The weighted-kernel DA enjoys the best of both worlds. On one hand, the algorithm is independent of initialization, and on the other hand, the method does not require to compute eigenvectors. Further, it is shown that by choosing the weights in particular ways, the weighted kernel DA objective function is *identical* to the normalized cut. This equivalence has an important implication: we can use DA-like iterative algorithms for directly minimizing the normalized-cut of a graph.

A word about notations. Capital letters such as A, X, Y and Φ denote matrices; lower-case bold letters such as $\boldsymbol{a}$, $\boldsymbol{b}$ denote column vectors; script-letters $\mathscr{A}, \mathscr{B}, \mathscr{V}$ and $\mathscr{E}$ represent sets; $\|\boldsymbol{a}\|$ denotes the L^2-norm of a vector; and $\|X\|_F$ denotes the Frobenius norm of a matrix, and is given by $\|X\|_F = \left(\sum_{i,j} X_{ij}^2\right)^2$.

The rest of the report is organized as follows. Section 2 introduces the basic DA algorithm by Rose, which is modified for shape-clustering applications using kernel trick in Section 3. We then specify kernel-DA's equivalence to spectral clustering in Section 4, followed by a couple of clustering examples in Section 5.

2 Deterministic Annealing (DA)

At its core, the Deterministic Annealing (DA) algorithm solves a facility location problem (FLP), i.e., given a set of demand point locations $\mathscr{X} = \{\boldsymbol{x_i}, i \in \mathbb{N}_{[1,N]}\}$, find $K \in \mathbb{N}$ facility locations $\mathscr{Y} = \{\boldsymbol{y_j}, j \in \mathbb{N}_{[1,K]}\}$ such that the *total weighted sum of the distance of each demand point from its nearest facility location is minimized.* Borrowing from the data compression literature, we define *distortion* as a measure of the average distance of a demand point to its nearest facility, given by $D(\mathscr{X}, \mathscr{Y}) = \sum\limits_{i \in \mathbb{N}_{[1,N]}} p(x_i) \min\limits_{j \in \mathbb{N}_{[1,K]}} d(x_i, y_j)$. The solution to an FLP satisfies the following two necessary (but not necessarily sufficient) properties:

- Voronoi partitions: The partition of the domain is such that each demand point in the domain is associated only to its nearest resource (cluster) location.

- Centroid condition: The resource location $\boldsymbol{y_j}$ is at the centroid of the j^{th} cluster C_j.

Most algorithms for FLP (such as Lloyd's [3]) are overly sensitive to the initial resource locations. This is primarily due to the distributed aspect of the FLPs, where any change in the location of the i^{th} demand point affects $d(\boldsymbol{x_i}, \boldsymbol{y_j})$ only with respect to the *nearest* facility j. The DA algorithm suggested by Rose [1], overcomes this sensitivity by allowing *fuzzy* association of every demand point to each facility through an association probability, $p(y_j|x_i)$:

$$\bar{D}(\mathscr{X},\mathscr{Y}) = \sum_{i\in\mathbb{N}_{[1,N]}} p(\boldsymbol{x_i}) \sum_{j\in\mathbb{N}_{[1,K]}} p(\boldsymbol{y_j}|\boldsymbol{x_i}) d(\boldsymbol{x_i}, \boldsymbol{y_j}). \tag{2.1}$$

Thus the notion of average distance of a demand point from its nearest facility is replaced by the weighted average distance of demand points to *all* the facilities. The probability distribution $\{p(y_j|x_i)\}$ determines the trade-off between decreasing the *local* influence and the deviation of the modified distortion $\bar{D}$ from the original distortion measure D. The uncertainties in facility locations $\{\boldsymbol{y_j}\}$ with respect to the demand point locations $\{\boldsymbol{x_i}\}$ is captured by Shannon *entropy* $H(\mathscr{Y}|\mathscr{X}) = -\sum_{i\in\mathbb{N}_{[1,N]}} p(\boldsymbol{x_i}) \sum_{j\in\mathbb{N}_{[1,K]}} p(\boldsymbol{y_j}|\boldsymbol{x_i}) \log(p(\boldsymbol{y_j}|\boldsymbol{x_i}))$, widely used in data compression literature. Therefore, maximizing the entropy is commensurate with decreasing the *local* influence.

This trade-off between maximizing the entropy and minimizing the distortion in Eq. (2.1) is addressed by seeking the probability distribution $\{p(\boldsymbol{y_j}|\boldsymbol{x_i})\}$ that minimize the *free-energy*, or the Lagrangian, given by $F := \bar{D}(\mathscr{X},\mathscr{Y}) - \frac{1}{\beta} H(\mathscr{Y}|\mathscr{X})$, where β is the Lagrange multiplier and bears a direct analogy to the inverse of the *temperature* variable in an annealing process [2]. The association weights $\{p(\boldsymbol{y_j}|\boldsymbol{x_i})\}$ that minimize the free-energy function are given by the *Gibbs* distribution

$$p(\boldsymbol{y_j}|\boldsymbol{x_i}) = \frac{e^{-\beta d(\boldsymbol{x_i},\boldsymbol{y_j})}}{\sum_{j\in\mathbb{N}_{[1,K]}} e^{-\beta d(\boldsymbol{x_i},\boldsymbol{y_j})}}. \tag{2.2}$$

By substituting the Gibbs distribution (2.2), the corresponding *free-energy* function is obtained as

$$F(\mathscr{Y}) = -\frac{1}{\beta} \sum_{i\in\mathbb{N}_{[1,N]}} p(\boldsymbol{x_i}) \log\Big(\sum_{j\in\mathbb{N}_{[1,K]}} e^{-\beta d(\boldsymbol{x_i},\boldsymbol{y_j})}\Big). \tag{2.3}$$

In the DA algorithm, the free-energy function is *deterministically* optimized at successively increased values of the annealing parameter β. Note that $d(\boldsymbol{x_i}, \boldsymbol{y_j}))$ is typically chosen as the squared-Euclidean distance, i.e.,

$$d(\boldsymbol{x_i}, \boldsymbol{y_j})) = \|x_i - y_j\|^2. \tag{2.4}$$

Taking derivative of the free-energy function w.r.t. y_j results in the following constraint equation

$$\boldsymbol{y_j} = \frac{\sum_i p(\boldsymbol{x_i}) p(\boldsymbol{y_j}|\boldsymbol{x_i}) \boldsymbol{x_i}}{\sum_i p(\boldsymbol{x_i}) p(\boldsymbol{y_j}|\boldsymbol{x_i})}. \tag{2.5}$$

The above equation has a form similar to computing centroid in k-means clustering algorithm. However, in k-means clustering, the association between $\boldsymbol{x_i}$ and $\boldsymbol{y_j}$ are hard (0-1). The DA algorithm alternates between Eqs. (2.2) and (2.5) at each β until convergence.

3 Weighted-kernel DA

The DA clustering algorithm can be enhanced by the use of a kernel function; by using an appropriate nonlinear mapping from the original (input) space to a higher dimensional feature space, one can extract clusters that are non-linearly separable in input space. Furthermore, we can generalize the kernel DA algorithm by introducing a specific choice of weight $p(\boldsymbol{x_i})$ for each point $\boldsymbol{x_i}$. As we shall see later, this generalization is powerful and encompasses the normalized cut of a graph.

Using the non-linear function ϕ, the distortion function of the weighted kernel-DA is defined as:

$$\bar{D}(\mathscr{X},\mathscr{Y}) = \sum_{i\in\mathbb{N}_{[1,N]}} p(\boldsymbol{x_i}) \sum_{j\in\mathbb{N}_{[1,K]}} p(\boldsymbol{y_j}|\phi(\boldsymbol{x_i}))\|\phi(\boldsymbol{x_i}) - \boldsymbol{y_j}\|^2, \tag{3.1}$$

where

$$\begin{aligned} \boldsymbol{y_j} &= \frac{\sum_i p(\boldsymbol{x_i})p(\boldsymbol{y_j}|\phi(\boldsymbol{x_i}))\phi(\boldsymbol{x_i})}{\sum_i p(\boldsymbol{x_i})p(\boldsymbol{y_j}|\phi(\boldsymbol{x_i}))}, \\ p(\boldsymbol{y_j}|\phi(\boldsymbol{x_i})) &= \frac{e^{-\beta\|\phi(\boldsymbol{x_i})-\boldsymbol{y_j}\|^2}}{\sum_{j\in\mathbb{N}_{[1,K]}} e^{-\beta\|\phi(\boldsymbol{x_i})-\boldsymbol{y_j}\|^2}}. \end{aligned} \tag{3.2}$$

Thus the Euclidean distance from $\phi(\boldsymbol{x_i})$ to centroid $\boldsymbol{y_j}$ is given by

$$\|\phi(\boldsymbol{x_i}) - \boldsymbol{y_j}\|^2 = < \phi(\boldsymbol{x_i}),\phi(\boldsymbol{x_i}) > + < \boldsymbol{y_j},\boldsymbol{y_j} > -2 < \phi(\boldsymbol{x_i}),\boldsymbol{y_j} > \tag{3.3}$$

The inner-products $< \phi(\boldsymbol{x_l}),\phi(\boldsymbol{x_m}) >$ are computed using kernel functions κ (e.g. Gaussian, polynomial or rbf-kernel), and are contained in the kernel matrix K. All computation in (3.3) is in the form of such inner products, hence we can replace all inner products by entries of the kernel matrix, i.e.,

$$\begin{aligned} \|\phi(\boldsymbol{x_i}) - \boldsymbol{y_j}\|^2 =& K_{ii} - 2\frac{\sum_{l=1}^N p(\boldsymbol{x_l})p(\boldsymbol{y_j}|\phi(\boldsymbol{x_l}))K_{il}}{\sum_{l=1}^N p(\boldsymbol{x_l})p(\boldsymbol{y_j}|\phi(\boldsymbol{x_l}))} \\ &+ \frac{\sum_{l,m=1}^N p(\boldsymbol{x_l})p(\boldsymbol{x_m})p(\boldsymbol{y_j}|\phi(\boldsymbol{x_l}))p(\boldsymbol{y_j}|\phi(\boldsymbol{x_m}))K_{lm}}{\left(\sum_{l=1}^N p(\boldsymbol{x_l})p(\boldsymbol{y_j}|\phi(\boldsymbol{x_l}))\right)^2} \end{aligned} \tag{3.4}$$

In the weighted-kernel DA algorithm, the Euclidean distance in (3.4) is iteratively computed until convergence at each β value.

4 Spectral connection

At first glance, weighted kernel DA and normalized cuts using spectral clustering appear to be quite different. After all, spectral clustering uses eigenvectors to help determine the partitions, whereas eigenvectors do not appear to figure in kernel DA. However, we know that the normalized cut problem can be expressed as a trace maximization problem, and in this section, we show how we can express weighted kernel DA as a trace maximization problem as well. This will show how to connect the two methods of clustering.

For further discussion, we make use of following notations. The distortion of cluster C_j is given by $D(C_j) = \sum_{i \in C_j} p(\boldsymbol{x_i}) \|\phi(\boldsymbol{x_i}) - \boldsymbol{y_j}\|^2$. Then the total distortion $\bar{D} = \sum_j D(C_j)$. Moreover, let us denote, for a cluster C_j, the sum of $p(\boldsymbol{x_i})$ weights of points in C_j to be s_j, i.e., $s_j = \sum_{i \in C_j} p(\boldsymbol{x_i})$. Finally, let us denote W to be the diagonal matrix of all the p weights, and W_j to be the diagonal matrix of the weights in C_j. Then we can rewrite the centroid $\boldsymbol{y_j}$ as

$$\boldsymbol{y_j} = \boldsymbol{\Phi_j} \frac{W_j \boldsymbol{e_j}}{s_j},$$

where $\boldsymbol{\Phi_j}$ is the matrix of points associated with cluster C_j, the full matrix of points $\Phi = [\boldsymbol{\Phi_1}, \dots, \boldsymbol{\Phi_k}]$, and $\boldsymbol{e_j}$ is the vector of all ones of appropriate sizes. It is shown in [6], that the minimization of total-distortion is equivalent to the following trace maximization problem, given by

$$\min_{\{C_j\}, \{\mathscr{Y}\}} \bar{D}(\phi(\mathscr{X}), \mathscr{Y}) \quad \equiv \quad \max_{Y} \text{Trace} \left(Y^T W^{1/2} \underbrace{\Phi^T \Phi}_{K} W^{1/2} Y \right), \tag{4.1}$$

where

$$Y = \begin{bmatrix} \frac{W_1^{1/2} \boldsymbol{e_1}}{\sqrt{s_1}} & & & \\ & \frac{W_2^{1/2} \boldsymbol{e_2}}{\sqrt{s_2}} & & \\ & & \cdots & \\ & & & \frac{W_k^{1/2} \boldsymbol{e_k}}{\sqrt{s_k}} \end{bmatrix}$$

Note that Y is an $N \times k$ orthonormal matrix, i.e., $Y^T Y = I$. A standard result in linear algebra provides a global solution to a relaxed version of this problem. By allowing Y to be an arbitrary orthonormal matrix, we can obtain an optimal Y by taking the top k eigenvectors of $W^{1/2} K W^{1/2}$. Similarly, the sum of the top k eigenvalues of $W^{1/2} K W^{1/2}$ gives the optimal trace value.

On the other hand, for a graph G with edge-weight matrix A and degree-matrix D, the optimization problem for the relaxed normalized cut problem is given by

$$\max_{Y} \text{Trace} \left(Y^T D^{-1/2} \underbrace{\Phi^T \Phi}_{A} D^{-1/2} Y \right) \qquad \text{s.t.} \qquad Y^T Y = I. \tag{4.2}$$

Note that the optimization problem in (4.1) is similar to optimization problem in (4.2). Thus, if we consider weighted kernel DA with $W = D$ and $K = D^{-1/2} A D^{-1/2}$, the two problems are identical. Thus, if the affinity matrix K is positive definite, we can use the weighted kernel DA procedure described above in order to minimize the normalized cut, without the need to compute eigenvectors.

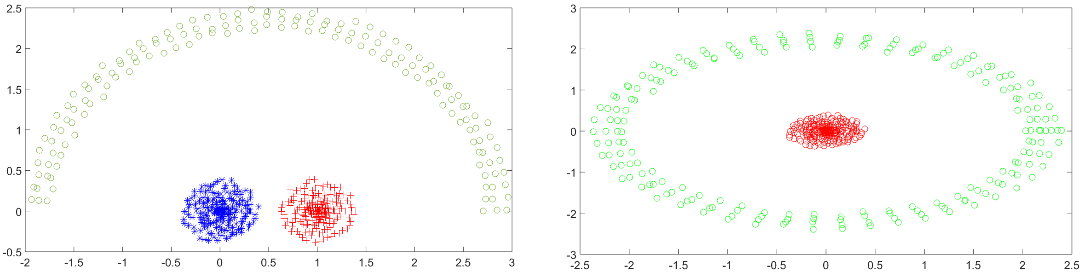

Figure 5.1: Results of kernel-DA approach for shape clustering

5 Results

We now describe some preliminary results of the kernel-DA method. We consider two instances - (1) semicircle and two disks, (2) concentric ring and a disk. Fig. 5 shows the performance of the proposed kernel-DA method for the two instances. As can be seen from the figure, the kernel-DA correctly identifies the underlying shapes. Note that a geometric scheduling rate of β update (i.e. $\beta_{t+1} = 1.05 * \beta_t$) is employed and thus results in fast clustering performance. The kernel matrices are generated using Gaussian kernels.

References

[1] Rose, Kenneth. "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems." Proceedings of the IEEE 86.11 (1998): 2210-2239.

[2] Jaynes, Edwin T. "Information theory and statistical mechanics." Physical review 106.4 (1957): 620.

[3] Lloyd, Stuart. "Least squares quantization in PCM." IEEE transactions on information theory 28.2 (1982): 129-137.

[4] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." IEEE Transactions on pattern analysis and machine intelligence 22.8 (2000): 888-905.

[5] Von Luxburg, Ulrike. "A tutorial on spectral clustering." Statistics and computing 17.4 (2007): 395-416.

[6] Dhillon, Inderjit S., Yuqiang Guan, and Brian Kulis. "Kernel k-means: spectral clustering and normalized cuts." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.