WEIGHTED KERNEL DETERMINISTIC ANNEALING

A Project Report

submitted by

K VAMSI KRISHNA

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **WEIGHTED KERNEL DETERMINISTIC ANNEALING**, submitted by **K Vamsi Krishna**, to the Indian Institute of Technology, Madras, for the award of the degree of **MASTER OF TECHNOLOGY**, is a bonafide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

—

Dr. Bharath Bhikkaji
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600036

Place: Chennai

Date: May 2018

ACKNOWLEDGEMENTS

I have put my best efforts into completion of this project. None of this would have been possible without the kind support and help of many individuals and organization. I would like to extend my sincere thanks to all of them.
I am grateful to my guide, Bharath Bhikkaji Sir, for having faith in my skills and providing me a project which aligned with my skills also the platform to demonstrate the same. It would not have been possible without the kind of support, feedback and help he provided to me throughout my project. The constant motivation from him helped me during the whole year.

ABSTRACT

K-Means and Spectral clustering methods are being extensively used for clustering purposes. K-means has limitations such as global optima problems and non-linear boundaries problems. Spectral methods can solve non-linear boundaries problem but they are difficult to compute because algorithms are based on finding eigen vectors. Deterministic Annealing algorithm is presented to solve the problem of global optima which is similar to the chemical process annealing, used to bring elements to their lower energy state. Weighted kernel deterministic annealing algorithm, which is a combination of kernel method and deterministic annealing, is presented to solve both global optima problem and non-linear boundary problems. Finally results of deterministic annealing and weighted kernel deterministic annealing on several interesting data sets are compared for visual understanding.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

DA	Deterministic Annealing
FLP	Facility Location Problem
WKDA	Weighted kernel Deterministic Annealing

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF MACHINE LEARNING

Machine learning is a branch of computer science that uses statistical techniques to provide systems the ability to learn from data, identify patterns and make decisions with less human intervention. Machine learning enables analysis of massive quantities of data. It can deliver faster, more accurate results in order to identify profitable opportunities or dangerous risks but may also require additional time and resources to train it properly. Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms are used when we need to predict the output from input using different data sets and their outputs. The goal is to train the system to learn the mapping function from input to the output with the help of given data sets. The system will be able to predict correct output for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly. Supervised learning can be further classified into Classification and Regression. If the output variable of the problem is a category, then the problem is classification problem. If the output variable of the problem is a value, then the problem is regression problem.

Unsupervised machine learning algorithms are used when the information used to train is unlabelled or not classified. The goal is not to figure out the right output, but to explore the data, draw inferences from data sets and describe hidden structures from the data to learn more about the data. Unsupervised learning studies how systems can create a function to describe a hidden structure from unlabelled data. Unsupervised learning can be further classified into Clustering and Association. If aim of the problem is to discover inherent grouping in the data, then the problem is clustering problem. If aim of the problem is to discover rules that describe large portions of the data, then the problem is association problem.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning since they use both labelled and unlabelled data for training. Typically, a small amount of labelled data and a large amount of unlabelled data is used for training. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it or learn from it. Otherwise, acquiring unlabelled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method in which the system learns by interacting with its environment by producing actions and observing the results as errors or rewards. This method allows to automatically determine the best action for a state in order to maximize its performance. If enough states are observed, an optimal decision policy can be generated for producing perfect actions in that particular environment.

1.2 CLUSTERING

Clustering is a task of dividing a set of data points into groups based on characteristics and similarities such that two points belonging to same group are more similar than two points belonging to different groups. These groups are called 'clusters'.

1.2.1 TYPES OF CLUSTERING METHODS

There are two types of clustering methods.
Hard Clustering: Hard clustering is a type of clustering in which either each data point strictly belongs to a cluster or it does not.
Soft Clustering: Soft clustering is a type of clustering which allows data points to part of more than one cluster with some probability.

1.2.2 TYPES OF CLUSTERING ALGORITHMS

Different algorithms use different methods to define similarity between two points. Some of them are mentioned below.

Centroid based: Similarity measure used in centroid based clustering algorithms is distance. Each data points belongs to the cluster to whose centroid it is closest to. In these type of algorithms numbers of clusters must be pre-defined. K-Means is the famous algorithm which is based on this method.

Connectivity based: These are similar to centroid based clustering algorithms. This type of clustering is also called hierarchical clustering. These algorithms does not give a single result but give hierarchy of results and one must choose appropriate result. There are two types in this type of clustering:

- **Agglomerative:** Each data point belong to its own cluster and clusters are merged as we iterate.
- **Divisive:** All data points belong to same cluster and divided in to different clusters as we iterate.

Different distance functions are used as similarity or dissimilarity measure to merge or split clusters.

Density based: In these type of clustering algorithms clusters are divided based on density of data points in data space. Different density areas are divided into different clusters.

Distribution based: In these type of clustering algorithms, data points are divided in such a way that points belonging to same cluster belong to similar distribution. Simply, different clusters have different distribitions.

1.3 K-MEANS CLUSTERING ALGORITHM

K-Means is one of the simplest unsupervised learning algorithms to solve well known clustering problems. The input data provided is unstructured or unlabelled. The aim of this algorithm is to organize the data in to groups or clusters. Here k denotes the number of clusters in the data. One of the key things is that we don't know how many clusters are there inside the provided data. So, what we do is to start with a minimum value for k and run the algorithm by increasing k. We stop once we hit an appropriate value for k. So, k becomes our decision point.

Initially k points are selected from the data. The points are called cluster centres or centroids. Let number of points in the data set be n. Let

$$X = \left\{x_1, x_2, x_3, ...x_n\right\} \tag{1.1}$$

represent the data points and

$$Y = \left\{y_1, y_2, y_3, ...y_k\right\} \tag{1.2}$$

represent the centroids. All data points are mapped to one of these centroids based on similarity measures. These similarity measures can be Euclidean distance, Manhattan distance, Bergman distance. In this algorithm we use Euclidean distance. So, a distance matrix D_{nxk}, containing distance from every data point to all centroids, is created. Every data point is assigned to its nearest centroid. Let

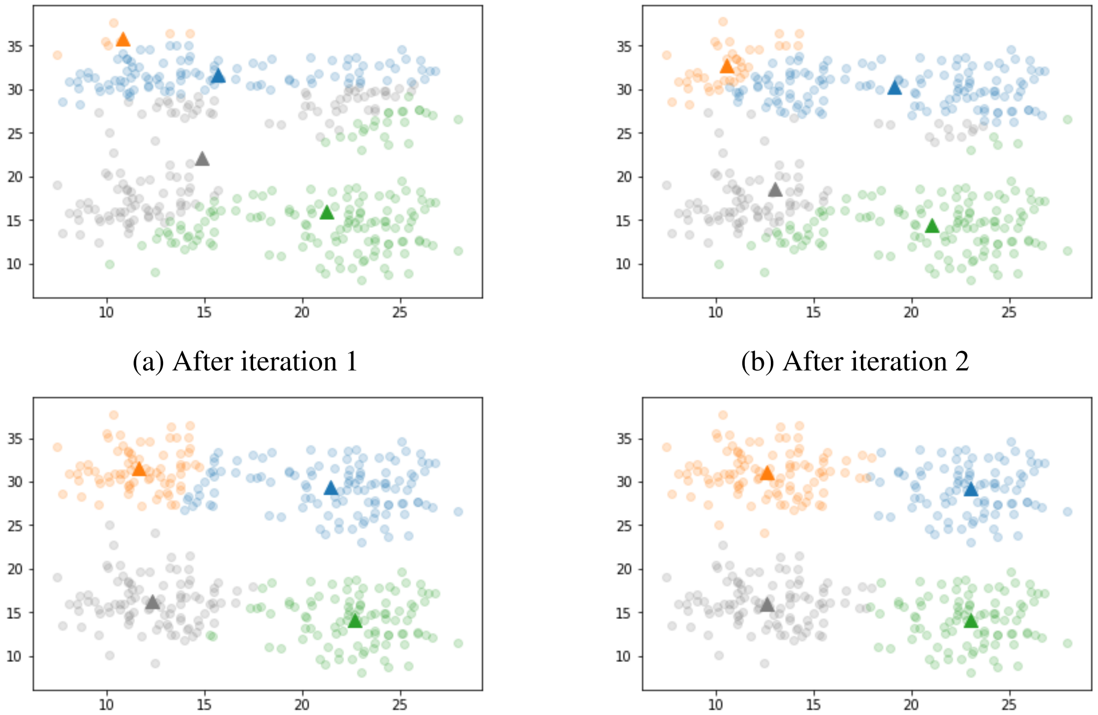
$$C = \left\{C_1, C_2, C_3, ...C_k\right\} \tag{1.3}$$

denote clusters. All points assigned to cluster i are represented by C_i. Centroids are recomputed by takings the mean of all points belonging to same cluster. Then, new centroids will be

$$y_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \qquad i = i, 2, ...n \quad j = 1, 2, ...k \tag{1.4}$$

where $|C_j|$ denotes number of points assigned to C_j. The above two steps (centroids are calculated and points are assigned to nearest centroids) are iterated till we see no change in centroids or maximum number of iterations are reached. All x_i assigned to a specified y_j form the cluster C_j.

K-Means algorithm is implemented on a data set with 4 clusters. Figures 1.1(a), 1.1(b), 1.1(c), 1.1(d) represent clustering results after iterations 1, 2, 3 and 7 respectively. Points belonging to different clusters are represented in different colours. Cluster cen-



(a) After iteration 1

(b) After iteration 2

(c) After iteration 3

(d) After iteration 7

Figure 1.1: Implementation of K-Means Algorithm

troids for different clusters are represented by dark and enlarged markers. From figure 1.1(a) we can see 4 random points are taken as cluster centroids. Figure 1.1(d) shows the result with four clusters separated with different markers. From the figures we can infer that, after every iteration centroid is moving closer to actual centroid.

1.3.1 ADVANTAGES AND DISADVANTAGES OF K-MEANS

K-Means algorithm is easier to implement and it takes less time to compute when compared with other clustering algorithms. K-Means algorithm works well if the data is linearly separable.

Initial centroids have great impact on results. If initial centroids are as in figure 1.2(a) then, the algorithm took 7 iterations to get to the result as in figure 1.2(b). If initial centroids are as in figure 1.2(c) then, the algorithm took 20 iterations to get to the result as in figure 1.2(d). Different initial centroids imply different number of iterations to get to the result.

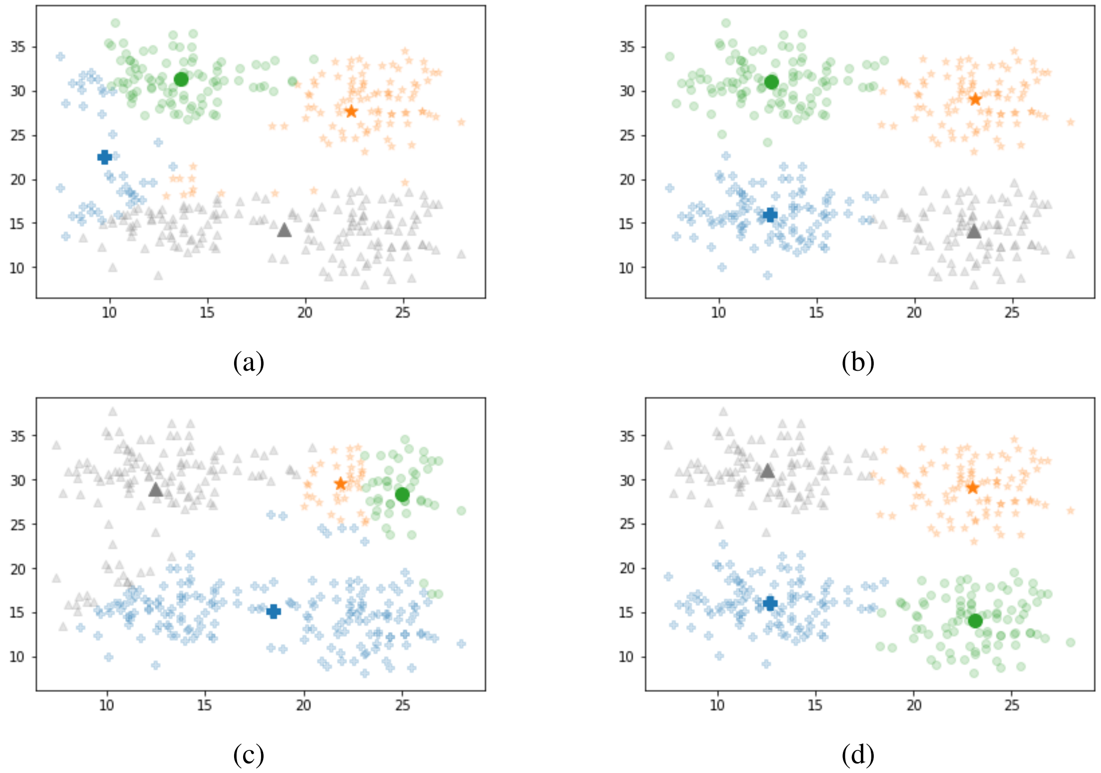


Figure 1.2: Dependence of number of iterations on initial centroids

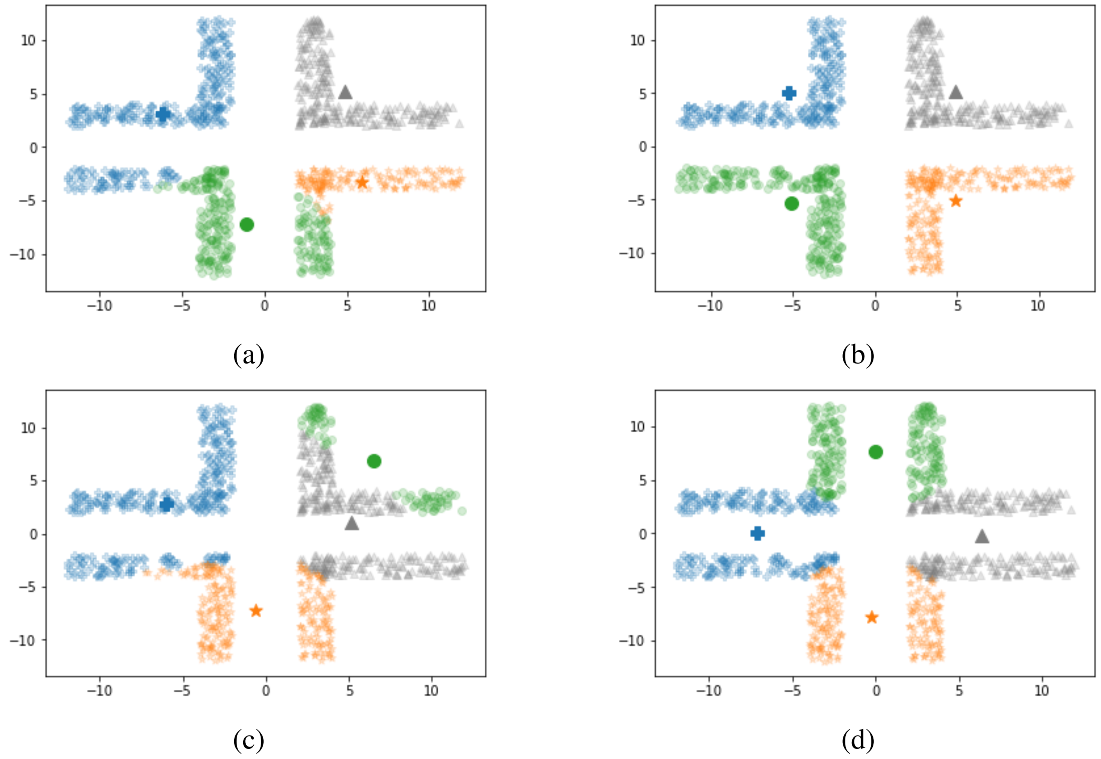
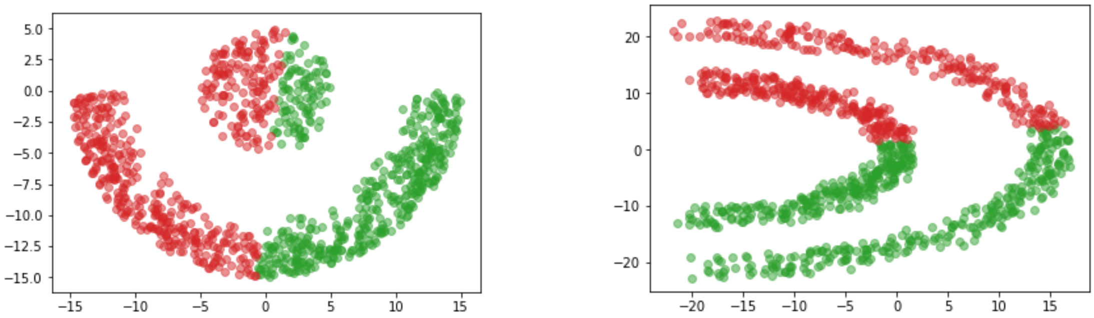


Figure 1.3: Dependence of global and local minimum on initial centroids

This algorithm will converge but that may not be the global minimum (it can be local minimum). Global minimum may be achieved by running the algorithm more number of times with randomized initial centroids. The data set shown in figures 1.1 and 1.2 always result in same output. If the clusters are in the form as shown in figure 1.3 we can observe that result is not always the same and it depends on initial centroid positions. If initial centroids are as in figure 1.3(a) then, the algorithm gives us the result as in figure 1.3(b). If initial centroids are as in figure 1.3(c) then, the algorithm gives us the result as in figure 1.3(d).

Distances from a cluster centroid to the data points belonging to that cluster are calculated (for all four clusters). Sum of all these distances is calculated. Sum is 3815.219 if the result is as shown in figure 1.3(b) which represents global minimum. Sum is 3981.147 if the result is as shown in figure 1.3(d) which represents local minimum. Even though the clusters are linearly separable, from figure 1.3(d) we can infer that the algorithm fails to separate the clusters. However, K-Means algorithm fails when the clusters are not linearly separable and figures 1.4(a) and 1.4(b) shows that.



(a)

(b)

Figure 1.4: Non-linearly separable clusters

1.3.2 K-MEANS ALGORITHM PSEUDO CODE

```
Select K random points as initial centroids
while Centroids do not change do
    Form K clusters by assigning each data point to its nearest centroid
    Recompute new centroids as mean of points belonging to same cluster
```

CHAPTER 2

DETERMINISTIC ANNEALING

2.1 INTRODUCTION

Resource allocation problems have large number of applications such as facility location problems, strategy planning, locational optimization. These type of problems are optimization problems whose cost function is non-convex with many local extrema. Due to non-convexity nature of these type of problems many optimization methods will result in local extrema but not global extrema which will change with change in initialization. We can solve this problem by checking with different initializations and taking the best extremum (maximum or minimum). But this is time consuming. Deterministic annealing solves this problem.

Annealing is a chemical process where a solid is heated till its melting point and cooled slowly at a determined rate to reach its lower energy state. In the corresponding probabilistic framework, a Gibbs distribution is defined over the set of all possible configurations which assigns higher probability to configuration of lower energy. This distribution is parameterized by the temperature, and as the temperature is lowered it becomes more discriminating (concentrating most of the probability in a smaller subset of low-energy configurations). At the limit of low temperature it assigns nonzero probability only to global minimum configurations. On the one hand it is deterministic, meaning that we do not want to be wandering randomly on the energy surface while making incremental progress on the average, as is the case for stochastic relaxation. On the other hand, it is still an annealing method and aims at the global minimum, instead of getting greedily attracted to a nearby local minimum.

2.2 DETERMINISTIC ANNEALING ALGORITHM

Deterministic Annealing is similar to facility location problem. Input data points are represented as $X = \{x_1, x_2, x_3, ...x_n\}$ which correspond to demand points in FLP. Cluster centers are represented as $Y = \{y_1, y_2, y_3, ...y_k\}$ which correspond to facility locations. Let clusters are denoted by $C = \{C_1, C_2, C_3, ...C_k\}$. Our goal is to find the optimal location of facilities so as to minimize transportation costs from set of points from their nearest facility locations. Weighted sum of distances of each data point from its nearest facility is given by

$$D(X, Y) = \sum_{j=1}^{k} \sum_{i=1}^{n} t_{ij} p(x_i) d(x_i, y_j) \tag{2.1}$$

where t_{ij} is association parameter ($t_{ij} = 1$ if y_j is the nearest facility to x_i and $t_{ij} = 1$ if it is not), $p(x_i)$ denotes relative significance and $d(x_i, y_j) =$ is distance x_i from y_j. Our algorithm is about clustering so, we take $p(x_i) = 1$ since every data point is equally significant. So, our objective function is to minimize $D(X, Y)$. $D(X, Y)$ is often referred as distortion between X and Y. Similar to many algorithms we need to take some random positions as initial positions for y_i and iterate. But this would result in local minima. So, we will solve this problem in a way similar to annealing.

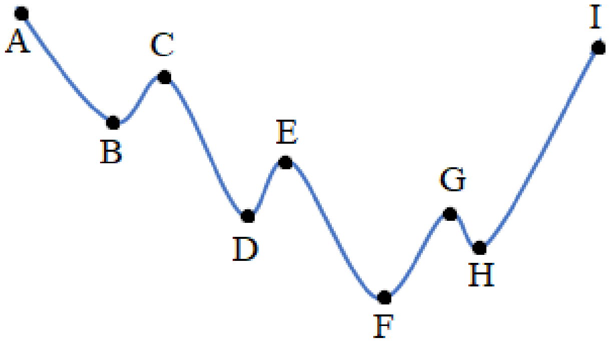


Figure 2.1: Curve with local and global minima

Consider the curve shown figure 2.1. Points B, D, H are local minima and point F is global minimum. If initial position does not fall between points E and G, it will always result local minimum if we simply minimize our objective function. According to many algorithms during iterations we move to a neighbour point which has minimum value for the function. By doing so we are never trying to cross the barriers BC, DE, HG to reach global minimum. So, we decide our next position non only by its functional value but also by some randomness which is controlled by temperature. Initially when the temperature is high, randomness will be high and it will be the dominating factor. Next point is decided randomly when T is high i.e., every point has equal probability to be the next point. When T is low functional value will be the dominating factor. Next point will be the neighbouring point which has the least functional value when T is low. By proceeding in this way there is a great probability to cross the barriers and reach global minimum. Now, $D\big(X,Y\big)$ changes as

$$D\big(X,Y\big) = \sum_{j=1}^{k}\sum_{i=1}^{n} p(y_j|x_i)d(x_i,y_j) \tag{2.2}$$

where $p(y_j|x_i$ are association probabilities. Level of randomness is measured by Shannon entropy

$$H\big(X,Y\big) = -\sum_{j=1}^{k}\sum_{i=1}^{n} p(x_i,y_j)log(p(x_i,y_j)) \tag{2.3}$$

Our objective function can be reformulated as minimization of lagrangian

$$F = D - TH \tag{2.4}$$

where T is the lagrange multiplier. For large values of T we are indirectly maximizing entropy and for small values of T we are directly minimizing D. Entropy can be further divided as: $H\big(X,Y\big) = H\big(X\big) + H\big(Y|X\big)$, where $H\big(X\big) = -\sum p(x)log(p(x))$ which is independent in the case of clustering and $H\big(Y|X\big) = -\sum\sum p(y|x)log(p(y|x)$. Now, F can be rewritten as

$$F = \sum_{j=1}^{k}\sum_{i=1}^{n} p(y_j|x_i)d(x_i,y_j) + T\sum_{j=1}^{k}\sum_{i=1}^{n} p(y_j|x_i)log(p(y_j|x_i) \tag{2.5}$$

Minimizing F with respect to $p(y|x)$ gives us the probability distribution for $p(y|x)$. It is Gibbs distribution.

$$p(y|x) = \frac{exp(-\frac{d(x,y)}{T})}{Z_x} \tag{2.6}$$

$$Z_x = \sum_{y} exp(-\frac{d(x,y)}{T}) \tag{2.7}$$

The corresponding minimum value for F is obtained by substituting equation (2.6) in equation (2.5)

$$F_{min} = min_{p(y|x)} F \tag{2.8}$$

$$F_{min} = -T \sum_x log \sum_y exp(-\frac{d(x,y)}{T}) \tag{2.9}$$

Minimizing F_{min}, for the squared error distortion measure, with respect to centroid locations gives us

$$y = \sum_x p(x|y)x = \frac{\sum_x p(x)p(y|x)x}{p(y)} = \frac{\sum_x p(x)p(y|x)x}{\sum_x p(y|x)p(x)} \tag{2.10}$$

$P(x) = 1$ in clustering problem. Therefore, y reduces to

$$y = \frac{\sum_x p(y|x)x}{\sum_x p(y|x)} \tag{2.11}$$

The algorithm is minimizing F_{min} with respect to y, starting with a high value for T and tracking down the minimum while lowering the value of T till it reaches a minimum value. Temperature is decreased in a determined way. If temperature is decreased slowly, time complexity of the algorithm increases which is not good. If temperature is decreased fast, we may miss the global minimum. The central iteration consists of two steps:

- Fix the centroid locations y_j and compute association probabilities $p(y_j|x_i)$ using equation (2.6)
- Fix the association probabilities $p(y_j|x_i)$ and compute centroid locations y_j using equation (2.11)

Above two steps are continuously iterated till the lagrangian F_{min} value doesn't change. Each and every x_i is assigned to a specified y_m such that $p(y_m|x_i) = max\{p(y_j|x_i)|j = 1, 2, ...k\}$. Generally when the temperature reaches the minimum value, the effect of shannon entropy (randomness) is negligible. Therefore, for a specified i, $p(y_j|x_i)$ will be 1 for a single value of j and will be 0 for remaining $k - 1$ values. So, we can say that x_i will be assigned to y_j if $p(y_j|x_i) = 1$. All x_i assigned to a specified y_j form the cluster C_j.

2.3 ADVANTAGES AND DISADVANTAGES Of DETERMINISTIC ANNEALING

The main advantage of deterministic annealing over remaining algorithms is its ability to attain global extrema. Deterministic annealing gives us a quality solution but its time complexity is bad compared to remaining algorithms. When the clusters are linearly separable DA algorithm works well. It fails when the clusters are non-linearly separable.

2.4 DETERMINISTIC ANNEALING PSEUDO CODE

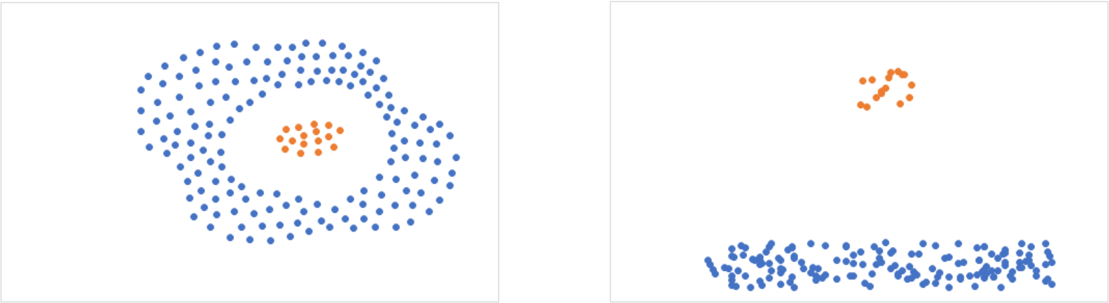
```
Select K random points as $y_j$
Set T to a big value
while Decreasing T till it reaches a minimum value do
    while $F_{min}$ doesn't change do
        Compute $p(y_j|x_i)$ using $y_j$
        Compute $y_j$ using $p(y_j|x_i)$
$x_i$ is assigned to $y_j$ if $p(y_j|x_i) = 1$ and all $x_i$ assigned to $y_j$ form cluster $C_j$
```

CHAPTER 3

WEIGHTED KERNEL DETERMINISTIC ANNEALING

3.1 INTRODUCTION

K-Means and Deterministic Annealing algorithms will not perform well when the clusters are non-linearly separable. Weighted Kernel Deterministic Annealing (WKDA) is used when the clusters are non-linearly separable. The idea is to project the input data, which is non-linearly separable, on to a higher dimensional feature space so that it is linearly separable in that feature space and then we perform deterministic annealing algorithm to divide the data into clusters. This method is called Kernel method. It helps us to use algorithms in higher dimensional feature space without actually calculating the coordinates of data points in that space and see the results in present lower dimensional space. Figures 3.1(a) and 3.1(b) represent data points in lower dimensional space and higher dimensional feature space respectively.



(a) (b)

Figure 3.1: Transformation of data points

Generally kernel function is of the form $K(x, y) =< \phi(x), \phi(y) >$. $\phi : X \to H$ is used to map the data point from lower dimensional space, X to higher dimensional feature space, H. To calculate $K(x, y)$ we need to calculate $\phi(x)$ and $\phi(y)$ and then do the dot product. This above process is a waste of resources and time because the function ϕ is difficult to calculate and $K(x, y)$ is a scalar. There is no point in going through all the trouble to get a number. So, we use pre defined kernel function. There are many types of many types of kernel functions. Some of which are:

- **Polynomial Kernel:** $K(x_i, x_j) = (x_i.x_j + 1)^d$ d is degree of polynomial
- **Gaussian Kernel:** $K(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$
- **Laplace RBF Kernel:** $K(x_i, x_j) = exp(-\frac{||x_i - x_j||}{\sigma})$
- **Hyperbolic Tangent Kernel:** $K(x_i, x_j) = tanh(k.x_i.x_j + c)$ k>0 and c<0

Gaussian Kernel is a general purpose kernel. We use it when we have no prior knowledge of data. For a function to be a kernel function it must satisfy mercer's condition. Mercer's condition states that a function is said to be a kernel function if and only if it is positive semi definite and symmetric.

3.2 WKDA ALGORITHM

Input data points are represented as $X = \{x_1, x_2, x_3, ...x_n\}$. Cluster centroids are represented as $Y = \{y_1, y_2, y_3, ...y_k\}$. Let clusters are denoted by $C = \{C_1, C_2, C_3, ...C_k\}$. Squared euclidean distance, $d(x_i, y_j)$ used in DA is replaced by $d(\phi(x_i), y_j)$. Since, we are computing the algorithm in higher dimensional feature space, we need to compute distance between $\phi(x_i)$ (data point in that space) and y_j. Squares euclidean distance can be represented as sum of inner products.

$$d(x_i, y_j) =< x_i, x_i > + < y_j, y_j > -2 < x_i, y_j > \tag{3.1}$$

$$d(\phi(x_i), y_j) =< \phi(x_i), x_i > + < y_j, y_j > -2 < \phi(x_i), y_j > \tag{3.2}$$

Kernel matrix K_{NxN} is computed using Gaussian kernel.

$$K_{ij} = K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{3.3}$$

DA algorithm with above mention changes is used to solve the clustering problem. So, similar to DA, lagrangian is calculated and minimized with respect to $p(y|x)$ to get probability distribution as

$$p(y|x) = \frac{exp(-\frac{d(\phi(x),y)}{T})}{Z_x} \tag{3.4}$$

$$Z_x = \sum_y exp(-\frac{d(\phi(x),y)}{T}) \tag{3.5}$$

F_{min} is calculated by substituting $p(y|x)$ in lagrange.

$$F_{min} = -T\sum_x log \sum_y exp(-\frac{d(\phi(x),y)}{T}) \tag{3.6}$$

Minimizing F_{min}, for the squared error distortion measure, with respect to centroid locations gives us

$$y = \frac{\sum_x p(y|x)\phi(x)}{\sum_x p(y|x)} \tag{3.7}$$

$d(\phi(x),y)$ in WKDA is given by

$$d(\phi(x_i),y_j) = K_{ii} - 2\frac{\sum_{l=1}^{n} p(y_j|x_l)K_{il}}{\sum_{l=1}^{n} p(y_j|x_l)} + \frac{\sum_{l=1}^{n}\sum_{m=1}^{n} p(y_j|x_l)p(y_j|x_m)K_{lm}}{\left(\sum_{l=1}^{n} p(y_j|x_l)\right)^2} \tag{3.8}$$

A distance matrix, D_{NxK} is created with $D_{ij} = d(\phi(x_i),y_j)$. Since we don't know $\phi(x)$ we cannot calculate centroid positions so, we iterate with distance function. Central iteration in WKDA consists of two steps:

- Fix association probabilities $p(y|x)$ and compute the distance matrix, D_{NxK} using equation (3.8)
- fix the distance matrix, D_{NxK} and compute association probabilities, $p(y|x)$ using equation (3.4)

3.3 ADVANTAGES AND DISADVANTAGES OF WKDA

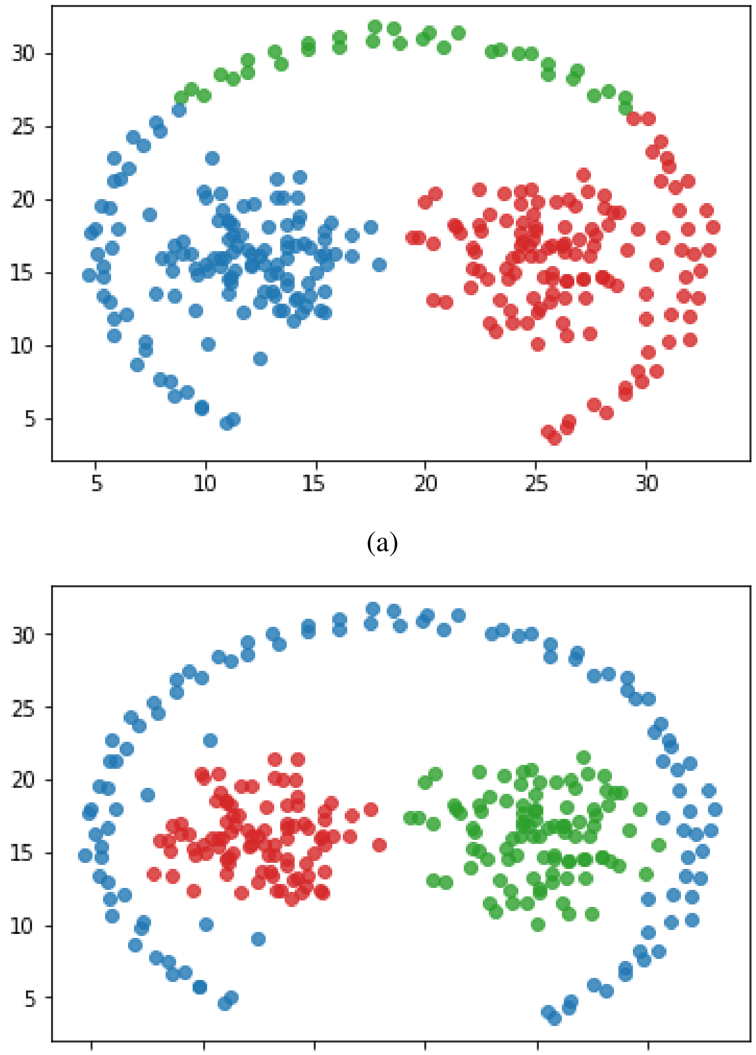
WKDA is one the best algorithms which can solve the problem of clustering when the clusters of non-linearly separable. Since DA is part of it, it promises us to give global optima. One of the disadvantages we can see in WKDA is its time complexity. Every iteration we need to calculate and store kernel matrix, K_{NxN} and distance matrix, D_{NxK}. We the data set contains clusters which are linearly separable, it is best not to use WKDA. To get best results in WKDA convergence limit and the value of sigma(σ) should be appropriately selected. The clustering results depend on sigma.

3.4 WKDA PSEUDO CODE

```
Assign random values for $p(y_j|x_i)$ such that $\sum_{j=1}^{k} p(y_j|x_i) = 1$ (or) we can simply take $p(y_j|x_i) = \frac{1}{k}$ for all i and j
Set T to a big value
while Decreasing T till it reaches a minimum value do
    while $F_{min}$ doesn't change do
        Compute $D_{ij}$ using $p(y_j|x_i)$
        Compute $p(y_j|x_i)$ using $D_{ij}$
$x_i$ is assigned to $y_j$ if $p(y_j|x_i) = 1$ and all $x_i$ assigned to $y_j$ form cluster $C_j$
```

CHAPTER 4

SIMULATION RESULTS

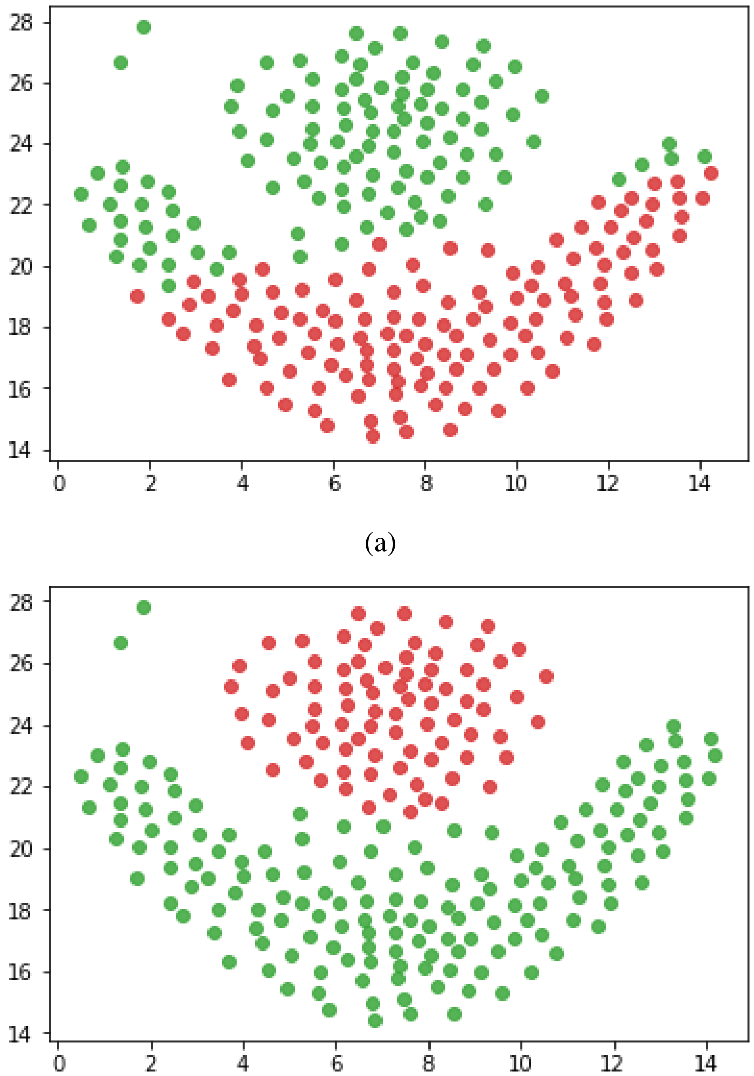


(a)

(b)

Figure 4.1: Difference of clustering results between DA and WKDA

In figure 4.1(a) DA couldn't separate the outer arc and the two inner clusters as different clusters since it cannot separate non-linearly separable clusters. In figure 4.1(b) WKDA was successful in separating them as different clusters. σ value for this data set is around 2.5-4.5.

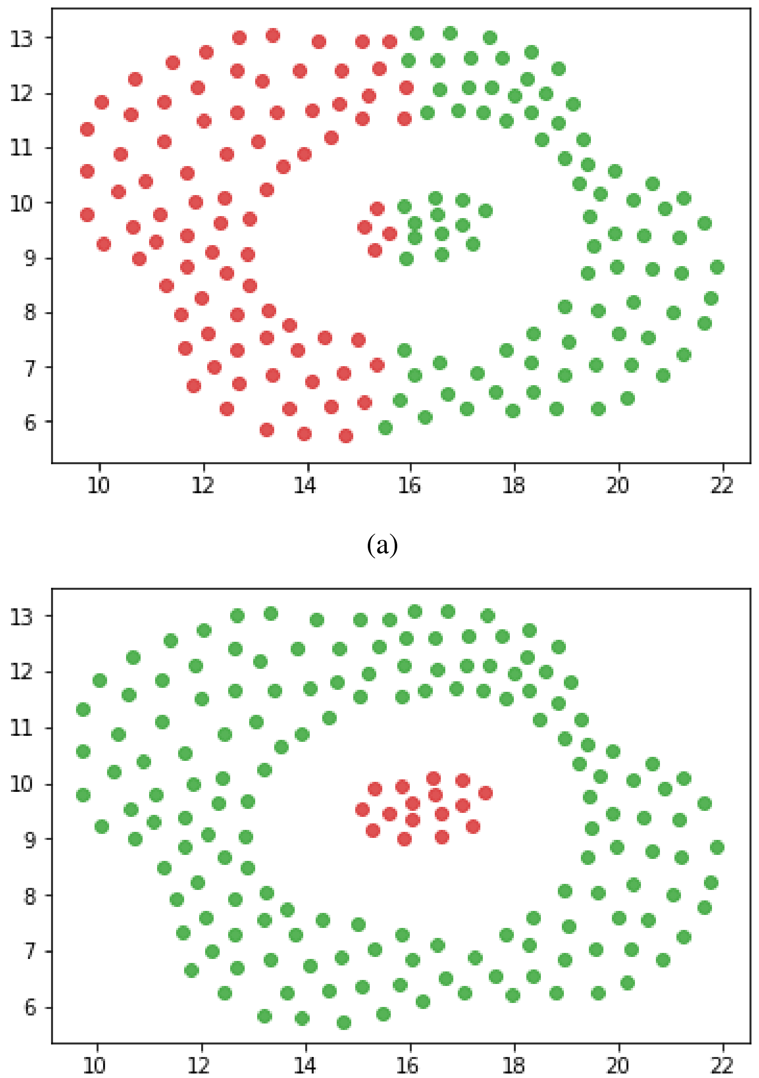


(a)

(b)

Figure 4.2: Difference of clustering results between DA and WKDA

In figure 4.2(a) DA couldn't separate upper and bottom parts of the data as two different clusters. Instead it separated them linearly. In figure 4.2(b) WKDA successfully separated them as two different clusters. σ value for this data set is around 1-3.



(a)

(b)

Figure 4.3: Difference of clustering results between DA and WKDA

In figure 4.3(a) DA couldn't find the outer cluster and inner cluster as two different clusters. It separated them linearly. In figure 4.3(b) WKDA successfully separated them without an error. σ value for this data set is around 0.3-0.5.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

’Weighted Kernel Deterministic Annealing’ is presented for shape clustering to solve for non-linearly separable data. Kernel method combined with ’Deterministic Annealing’ is used in the algorithm to get best results possible. WKDA is the best algorithm to use if the boundaries of clusters are non-linear. Here, in this algorithm we must trade time complexity for the quality of the solution. Time Complexity is huge compared to DA.

In future work, we can implement a more simpler version of WKDA with better rum time complexity. Based on clustering results similar type of photos can be grouped or compared. The portrait mode effect used in smart phones which is done using two cameras can be done with single camera with the help WKDA as all real life pictures are made of clusters with non-linear boundaries.

CHAPTER 6

REFERENCES

1. Mayank Baranwal and Srinivasa M. Salapaka, "Weighted Kernel Deterministic Annealing: A Maximum-Entropy Principle Approach for Shape Clustering", Indian Control Conference (ICC), 2018

2. K. Rose, “Deterministic annealing for clustering, compression, classification, regression, and related optimization problems,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2210–2239, 1998.

3. B. Kulis, S. Basu, I. Dhillon, and R. Mooney, “Semi-supervised graph clustering: a kernel approach,” in Proceedings of the 22nd international conference on Machine learning. ACM, 2005, pp. 457–464.

4. I. S. Dhillon, Y. Guan, and B. Kulis, “Kernel k-means: spectral clustering and normalized cuts,” in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004, pp. 551–556.

5. Some of the data sets are obtained from *https://cs.joensuu.fi/sipu/datasets/*