Automated Clustering of High-dimensional Data with a Feature Weighted Mean Shift Algorithm

Saptarshi Chakraborty[1*], **Debolina Paul**[2*], **Swagatam Das**[3†]
[1] Department of Statistics, University of California, Berkeley
[2] Indian Statistical Institute, Kolkata
[3] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata
saptarshic@berkeley.edu, debolinap8@gmail.com, swagatam.das@isical.ac.in

Abstract

Mean shift is a simple interactive procedure that gradually shifts data points towards the mode which denotes the highest density of data points in the region. Mean shift algorithms have been effectively used for data denoising, mode seeking, and finding the number of clusters in a dataset in an automated fashion. However, the merits of mean shift quickly fade away as the data dimensions increase and only a handful of features contain useful information about the cluster structure of the data. We propose a simple yet elegant feature-weighted variant of mean shift to efficiently learn the feature importance and thus, extending the merits of mean shift to high-dimensional data. The resulting algorithm not only outperforms the conventional mean shift clustering procedure but also preserves its computational simplicity. In addition, the proposed method comes with rigorous theoretical convergence guarantees and a convergence rate of at least a cubic order. The efficacy of our proposal is thoroughly assessed through experimental comparison against baseline and state-of-the-art clustering methods on synthetic as well as real-world datasets.

Introduction

Clustering, a cornerstone of unsupervised learning, refers to the task of partitioning a dataset into more than one exhaustive and mutually exclusive groups, based on some measure of similarity (Xu and Tian 2015). Some popular paradigms in clustering include center-based approaches such as k-means and its variants (Jain 2010), hierarchical clustering (Carlsson and Mémoli 2010), spectral clustering (Ng, Jordan, and Weiss 2002; Hess et al. 2019), density-based methods (Ester et al. 1996), convex clustering (Chi and Lange 2015), kernel clustering (Dhillon, Guan, and Kulis 2004), model-based frequentist approaches (McNicholas 2016) and Bayesian methods (Archambeau and Verleysen 2007; Kulis and Jordan 2012).

Most of the aforementioned algorithms inherently use the number of clusters (k) as an input. However, for real-world data, k may not be known beforehand. Determining k from the dataset itself has long been an open problem and has attracted a lot of attention from the relevant research community (Tibshirani, Walther, and Hastie 2001; Hamerly and Elkan 2004; Fischer 2011; Kulis and Jordan 2012; Chakraborty and Das 2018; Gupta, Datta, and Das 2018; Paul and Das 2020).

Moreover, algorithms for solving k-means type non-convex clustering problems are prone to get stuck at local minima (Xu and Lange 2019). Recent attempts to mitigate this issue approach the problem via annealing with a class of functions approximating the k-means type objective (Xu and Lange 2019; Chakraborty et al. 2020) or by taking a convex relaxation of the problem (Chi and Lange 2015; Pelckmans et al. 2005; Wang et al. 2018). There are also density-based algorithms like DBSCAN (Ester et al. 1996; Jiang 2017) or other mode seeking approaches like quick shift (Vedaldi and Soatto 2008; Jiang 2017) which attempts to speed up mean shift. However, these methods either require k as an input or their performance degrade in a high-dimensional setting, where the signal-to-noise-ratio is quite low.

To find the number of clusters automatically and to learn various properties of the feature space, researchers have resorted to the mean shift (MS) paradigm (Cheng 1995; Su and Shang 2017). Mean shift has previously been used for mode seeking, object tracking, and automated clustering in the feature space.

Suppose $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subset \mathbb{R}^p$ be n data points to be clustered. The mean shift initiates n points $\boldsymbol{y}_1^{(0)}, \dots, \boldsymbol{y}_n^{(0)}$ and updates $\boldsymbol{y}_i$ according to the following update rule:

$$\boldsymbol{y}_i^{(t+1)} = \frac{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{x}_j\|/h)\boldsymbol{x}_j}{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{x}_j\|/h)}, \tag{1}$$

until convergence. Here $K(\cdot)$ is a kernel function (e.g. the Gaussian kernel, $K(x) = \exp\{-x^2\}$) and h is the bandwidth parameter. This version of the mean shift is often used to detect the mode(s) of the estimated density. Often, instead of updating based on the original data points $\boldsymbol{x}_i$'s, one uses $\boldsymbol{y}_i^{(t)}$'s, the points obtained in the t-th step as follows:

$$\boldsymbol{y}_i^{(t+1)} = \frac{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|/h)\boldsymbol{y}_j^{(t)}}{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|/h)}. \tag{2}$$

[*] Co-first authors contributed equally
[†] Corresponding author

As before, $y_i^{(0)}$ is initiated at $\boldsymbol{x}_i$. This version of mean shift is referred to as Blurring Mean Shift (BMS). The updates in equation (2) can be thought of as putting a low-pass filter on the data and thus "blurring" out irregularities in the data. Apart from clustering, BMS has been used for data denoising and manifold learning (Wang and Carreira-Perpinán 2010).

Despite their simplicity, both the usual and blurring mean shift perform poorly for high-dimensional data. This is primarily because of the use of Euclidean distance, which becomes less informative as the number of feature increases due to the curse of dimensionality (Donoho et al. 2000). High-dimensional datasets often contain only a few relevant/ discriminating features, along with a huge number of irrelevant/noisy features, which severely affect the performance of mean shift and other popular clustering algorithms.

There is a rich literature on clustering high-dimensional data including subspace clustering (Kriegel, Kröger, and Zimek 2009; Elhamifar and Vidal 2013), bi-clustering (Chi, Allen, and Baraniuk 2017), dimensionality reduction based approaches (Jin, Wang et al. 2016) and data-depth based approaches (Sarkar and Ghosh 2019). However, most of these methods are computationally expensive. Towards finding efficient feature representation of high-dimensional data while clustering, weighted k-means (Huang et al. 2005) and Sparse k-means (Witten and Tibshirani 2010) have become benchmark algorithms for learning effective feature representations of such data. However, these methods also require k as input and thus, lose their appeal to the practitioner who may not have handled the data before and wants to find the number of clusters in an unsupervised manner as well.

This paper aims to develop a blurring mean shift based algorithm, which can automatically find an efficient feature representation of the data as well as the number of clusters *simultaneously*. Called the Weighted Blurring Mean Shift (WBMS), we extend the merits of blurring mean shift to high-dimensional data, while preserving its computational cost. To achieve this, we introduce a feature weight vector to learn the importance of each feature as the data is smoothed. The resulting iterations lead to an elegant and simple algorithm with closed-form updates. The weight update scheme follows the philosophy that features with higher within-cluster variance contribute less in finding the cluster structure of the data (Huang et al. 2005; Witten and Tibshirani 2010; Chakraborty et al. 2020). The main contributions of this paper can be summarized as follows:

- We introduce the Weighted Blurring Mean Shift (WBMS) formulation as an intuitive extension of mean shift to high-dimensional data clustering. This simple formulation is found to be effective in finding out the number of clusters and also filter out the unimportant features from the data *simultaneously*.
- The obtained feature weights can be interpreted as the outcome of an entropy regularization on the within-cluster sum of squares.
- The WBMS algorithm comes with closed-form updates and its convergence guarantees are discussed.
- We also analyze asymptotic convergence properties of the data cloud under the WBMS algorithm. We analytically show that the convergence of the spread of the data cloud is at least of cubic order.
- Through detailed experimental analysis, we show the efficacy of our proposed algorithm against state-of-the-art clustering techniques on a suit of simulated and real-world data. Our experimental results indicate that WBMS is especially effective, compared to its competitors, in a high-dimensional setting despite the low signal-to-noise-ratio.

Before proceeding to the details of WBMS, let us now present a motivating example demonstrating its potential.

A Motivating Example We generate a 200×32 dimensional data, called *data1*, which consists of two clusters with 100 points each. The cluster structure of *data1* is fully contained in the first two features and the rest of the 30 features are independently generated from a standard normal distribution, that contain no clustering information. We standardize (z-transform) the data before use and set the bandwidth $h = 0.1$. In Fig. 1, we show the position of the data cloud, plotted in the first two relevant feature dimensions, for both BMS and WBMS as the number of iterations (t) is increased. It can be easily seen that since BMS is more influenced by the combination of 30 Gaussian features, the data cloud gradually converges to the origin. On the other hand, the WBMS correctly identifies the two important and informative features. Thereby using this information, the data cloud converges to the two cluster centroids. The estimated density of the original data cloud $\mathcal{X}$, $\hat{f}(\boldsymbol{x}) \propto \sum_{i=1}^n \exp\{-\|\boldsymbol{x}-\boldsymbol{x}_i\|_{\boldsymbol{w}}^2/h\}$, with $\boldsymbol{w}$ being the feature weights found out by WBMS (see the following section for the definition of $\|\cdot\|_{\boldsymbol{w}}$), also closely resembles the density of the data without the presence of the 30 non-informative features.

Weighted Blurring Means Shift

In this section, we formulate the Weighted Blurring Mean Shift (WBMS) algorithm and discuss some of its intriguing properties. Throughout this paper, $\mathcal{N}_p(\boldsymbol{\mu}, \Sigma)$ denotes the p-variate normal distribution with mean $\boldsymbol{\mu}$ and dispersion matrix Σ. $Unif(A)$ denotes the uniform distribution over the set A.

Motivation

Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$ be n data points to be clustered. We note that the blurring mean shift updates (equation (2)) uses the Euclidean distance $\|\boldsymbol{x}-\boldsymbol{y}\|_2 = \sqrt{\sum_{l=1}^p (x_l - y_l)^2}$. The Euclidean distance puts equal weight on each of the components $(x_l - y_l)^2$ and thus, not suitable when there are many noisy features, which are irrelevant to the clustering of the data. Recent research (Witten and Tibshirani 2010; Chakraborty et al. 2020) has been focused in replacing the usual Euclidean distance with the weighted distance $\|\boldsymbol{x}-\boldsymbol{y}\|_{\boldsymbol{w}} = \sqrt{\sum_{l=1}^p w_l (x_l - y_l)^2}$. Here $w_l \geq 0$ denotes the feature weight of the l-th feature. It can be easily checked that $\|\cdot\|_{\boldsymbol{w}}$ defines a norm on $\mathbb{R}^p$. A large feature weight, w_l on the l-th feature gives more importance to the difference

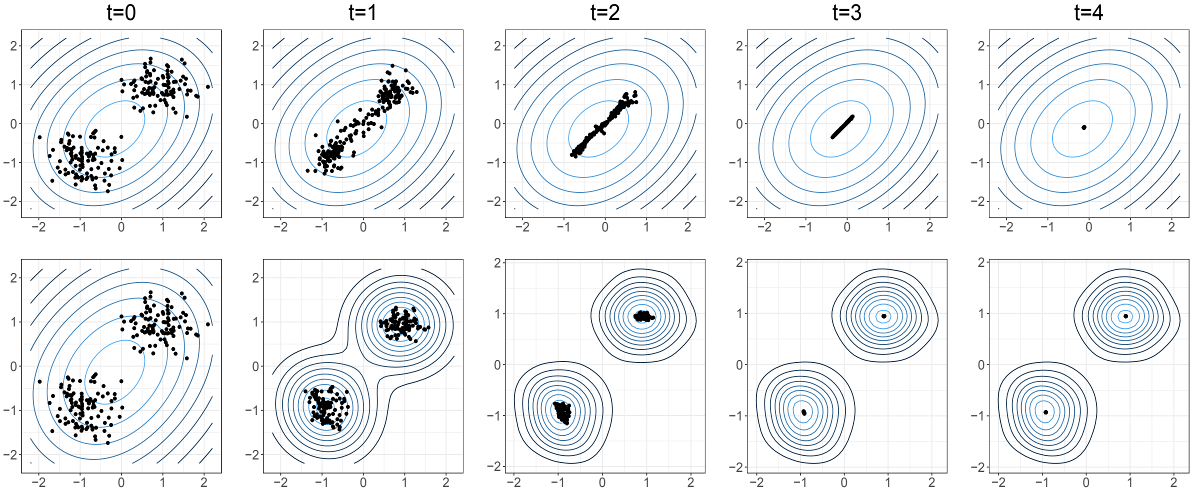

Figure 1: Performance of BMS (first row) and WBMS (second row) along with the contour plots of the estimated kernel density for the motivating example as the number of iterations (t) is increased. WBMS correctly identifies the two true cluster centroids and efficiently selects the relevant features, while the BMS fails to do so.

$|x_l - y_l|$, the discrimination between $\boldsymbol{x}$ and $\boldsymbol{y}$ along the l-th coordinate vector. $\boldsymbol{w}$ is called the feature weight vector and is usually normalized, i.e. $\mathbf{1}^\top \boldsymbol{w} = 1$. The feature weights are typically learned from the data and are updated every passing iteration. Normally, the feature weights are taken as some decreasing function of the within cluster sum of squares for that feature.

Formulation

We will use a similar update rule as in BMS (equation (2)). However, instead of the usual Euclidean distance, we will use the weighted distance $\| \cdot \|_{\boldsymbol{w}}$. The update rule for the data points is given by,

$$\boldsymbol{y}_i^{(t+1)} = \frac{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}/h)\boldsymbol{y}_j^{(t)}}{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}/h)}. \tag{3}$$

The feature weights are updated as follows:

$$w_l^{(t)} = \frac{\exp\{-\frac{1}{n\lambda}\sum_{i=1}^n (x_{il} - y_{il}^{(t)})^2\}}{\sum_{l'=1}^p \exp\{-\frac{1}{n\lambda}\sum_{i=1}^n (x_{il'} - y_{il'}^{(t)})^2\}}. \tag{4}$$

Algorithm 1 gives a formal description of the weighted blurring mean shift algorithm.

Detecting the Clusters

We observe that Algorithm 1 outputs $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, which are proxies for the cluster centroids. One should note that if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ were originally in the same cluster, then $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ should be close to each other in the Euclidean sense, i.e. $\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$ should be small enough. We construct an undirected graph with the adjacency matrix $\boldsymbol{A} = ((a_{ij}))$. For a prefixed tolerance ϵ (in our experiments, we take $\epsilon = 10^{-5}$), we will take,

$$a_{ij} = \begin{cases} 1, & \text{if } \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2 < \epsilon \\ 0, & \text{Otherwise.} \end{cases}$$

Algorithm 1 Weighted Blurring Mean Shift (WBMS) Algorithm

```
Input: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$, $h, \lambda > 0$
Output: $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ and $\boldsymbol{w}$.
Initialize $\boldsymbol{y}_i^{(0)} \leftarrow \boldsymbol{x}_i$ for all $i = 1, \ldots, n$.
Initialize $w_l^{(0)} = \frac{1}{p}$, for all $l = 1, \ldots, p$.
repeat
    Step 1: Update $\boldsymbol{y}_i$'s by,
        $\boldsymbol{y}_i^{(t+1)} \leftarrow \frac{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}/h)\boldsymbol{y}_j^{(t)}}{\sum_{j=1}^n K(\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}/h)}.$
    Step 2: Update $\boldsymbol{w}$ by,
        $w_l^{(t)} = \frac{\exp\{-\frac{1}{n\lambda}\sum_{i=1}^n (x_{il} - y_{il}^{(t)})^2\}}{\sum_{l'=1}^p \exp\{-\frac{1}{n\lambda}\sum_{i=1}^n (x_{il'} - y_{il'}^{(t)})^2\}}.$
until $\left| \max_{i,j} \|y_i^{(t+1)} - y_j^{(t+1)}\|_2 - \max_{i,j} \|y_i^{(t)} - y_j^{(t)}\|_2 \right|$ converges
```

Let $\mathcal{G}$ be a graph based on the adjacency matrix $\boldsymbol{A}$ and vertices $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. The clusters in $\mathcal{X}$ should ideally correspond to the connected components of $\mathcal{G}$. The number of clusters, k, corresponds to the number of connected components of $\mathcal{G}$. The connected components of $\mathcal{G}$ can easily be found by a Depth First Search or a Breadth First Search.

Implicit Entropy Regularization

We now show that the weight update scheme presented in equation (4) can be thought of as the outcome of an entropy regularization. At convergence $y_i^{(t)}$ can be treated as the cluster centroid corresponding to $\boldsymbol{x}_i$. Thus, the within cluster sum of squares is given by, $\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i - \boldsymbol{y}_i^{(t)}\|_{\boldsymbol{w}}^2$. Since we are imposing the constraint $\sum_{l=1}^{p} w_l = 1$, minimization of this within cluster sum of squares will result in a trivial coordinate vector of $\mathbb{R}^p$. As observed by (Chakraborty et al. 2020), this problem can be successfully avoided by adding an entropy incentive term as:

$$\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i - \boldsymbol{y}_i^{(t)}\|_{\boldsymbol{w}}^2 + \lambda\sum_{l=1}^{p} w_l \log w_l, \tag{5}$$

where $\lambda > 0$. Note that the second term in the above expression is the negative of Shannon's entropy of $\boldsymbol{w}$. Minimizing (5) subject to $\boldsymbol{w}^\top\boldsymbol{1} = 1$, results in the weight update formula, given in equation (4). This is assured by the following theorem.

Theorem 1. *Let $\boldsymbol{w}^*$ be the minimizer of (5), subject to $\boldsymbol{w}^\top\boldsymbol{1} = 1$. Then,*

$$w_l^* = \frac{\exp\{-\frac{1}{n\lambda}\sum_{i=1}^{n}(x_{il} - y_{il}^{(t)})^2\}}{\sum_{l'=1}^{p}\exp\{-\frac{1}{n\lambda}\sum_{i=1}^{n}(x_{il'} - y_{il'}^{(t)})^2\}}.$$

Theoretical Properties and Convergence Guarantees

We will now discuss some of the interesting properties of WBMS. In particular, we prove that WBMS converges after a finite number of iterations for any fixed tolerance. We also find the asymptotic rate of convergence of the cluster variance as the number of points. Some related theoretical work in this area can be found in (Carreira-Perpiñán 2006; Chen 2015; Huang, Fu, and Sidiropoulos 2018; Rocha et al. 2020). All the proofs pertaining to this section can be found in the supplement.

Convergence Guarantee

In this section, we will discuss some of the theoretical properties of the WBMS algorithm. For any set $A \subseteq \mathbb{R}^p$, let $\mathcal{C}(A)$ denote the convex hull of A. We begin our analysis by proving that the convex hulls of $\{\boldsymbol{y}_1^{(t)}, \ldots, \boldsymbol{y}_n^{(t)}\}$ for a decreasing sequence of sets in Theorem 2.

Theorem 2. *Let $C_t = \mathcal{C}(\{\boldsymbol{y}_1^{(t)}, \ldots, \boldsymbol{y}_n^{(t)}\})$. Then $\{C_t\}_{t=0}^{\infty}$ constitutes a decreasing sequence of sets, i.e.*

$$C_0 \supseteq C_1 \supseteq \ldots C_t \supseteq C_{t+1} \supseteq \ldots$$

Since $\{C_t\}_{t=0}^{\infty}$ forms a decreasing sequence of sets, we immediately get that C_t converges in the following corollary.

Corollary 1. *$\lim_{t\to\infty} C_t$ exists and is given by, $\lim_{t\to\infty} C_t = \cap_{t=1}^{\infty} C_t$.*

In Theorem 3, we derive that for any fixed tolerance, the convergence criterion of Algorithm 1 is satisfied after a number of finite iterations.

Theorem 3. *For any pre-fixed tolerance level δ, there exists $T \in \mathbb{N}$ such that*

$$\big|\max_{i,j}\|y_i^{(t+1)} - y_j^{(t+1)}\|_2 - \max_{i,j}\|y_i^{(t)} - y_j^{(t)}\|_2\big| < \delta,$$

for all $t \geq T$.

Convergence Rate

Let us now discuss the behavior of a Gaussian cluster under WBMS. We will show that the Gaussian cluster shrinks towards its mean with at least a cubic convergence rate. Let $\phi(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the Gaussian probability density function with mean $\boldsymbol{\mu}$ and dispersion matrix $\boldsymbol{\Sigma}$. In order to remove the dependency on the random process, we take an infinite sample, distributed in the whole of $\mathbb{R}^p$ according to the density $q(\boldsymbol{x})$. For simplicity, we consider the Gaussian kernel. The kernel density estimate at $\boldsymbol{z}$, based on the data $\boldsymbol{y}_1^{(t)}, \ldots, \boldsymbol{y}_n^{(t)}$ is given by,

$$\hat{p}_t(\boldsymbol{z}) = \frac{c(h, \boldsymbol{w}^{(t)})}{n}\sum_{j=1}^{n}\exp\big\{-\|\boldsymbol{z} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}^2/h\big\}.$$

Here $c(h, \boldsymbol{w}^{(t)})$ is a constant depending only on h and $\boldsymbol{w}^{(t)}$. From equation (3), we get,

$$\begin{aligned}
\boldsymbol{y}_i^{(t+1)} &= \frac{\sum_{j=1}^{n}\exp\{-\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}^2/h\}\boldsymbol{y}_j^{(t)}}{\sum_{j=1}^{n}\exp\{-\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}^2/h\}} \\
&= \frac{c(h, \boldsymbol{w}^{(t)})}{n}\sum_{j=1}^{n}\frac{e^{-\frac{1}{h}\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}^2}\boldsymbol{y}_j^{(t)}}{\frac{c(h,\boldsymbol{w}^{(t)})}{n}\sum\limits_{j=1}^{n} e^{-\frac{1}{h}\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}^2}} \\
&= \frac{c(h, \boldsymbol{w}^{(t)})}{n}\sum_{j=1}^{n}\frac{\exp\{-\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}_j^{(t)}\|_{\boldsymbol{w}^{(t)}}^2/h\}\boldsymbol{y}_j^{(t)}}{\hat{p}_t(\boldsymbol{y}_i^{(t)})} \\
&\approx c(h, \boldsymbol{w}^{(t)})\int \boldsymbol{y}\frac{\exp\{-\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}\|_{\boldsymbol{w}^{(t)}}^2/h\}}{p_t(\boldsymbol{y}_i^{(t)})}q_t(\boldsymbol{y})d\boldsymbol{y} \\
&= \int \boldsymbol{y}(p_t(\boldsymbol{y}_i^{(t)}))^{-1}\phi(\boldsymbol{y}_i^{(t)} - \boldsymbol{y}; 0, \frac{h}{2}diag(\frac{1}{\boldsymbol{w}^{(t)}}))q_t(\boldsymbol{y})d\boldsymbol{y}.
\end{aligned}$$

Thus, if the number of samples is large, each data point $\boldsymbol{z}$ is replaced by the conditional expectation $E(\boldsymbol{y}|\boldsymbol{z}) = \int \boldsymbol{y}p_t(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{y}$, where, $p_t(\boldsymbol{y}|\boldsymbol{z}) = \phi(\boldsymbol{z} - \boldsymbol{y}; 0, \frac{h}{2}diag(1/\boldsymbol{w}^{(t)}))q_t(\boldsymbol{y})/p_t(\boldsymbol{z})$. For simplicity of exposition, we begin with $\boldsymbol{x}$, which follows a Gaussian distribution with mean $\boldsymbol{0}$ and dispersion matrix $\boldsymbol{\Sigma} = diag(\sigma_1^2, \ldots, \sigma_p^2)$. From the above analysis, it is clear that at a population level, the WBMS can be thought of as taking consecutive conditional expectations w.r.t $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. Here $\boldsymbol{x}_t$ denote the population at the t-th step of the algorithm with $\boldsymbol{x}_0 = \boldsymbol{x}$. The following theorem asserts that $\boldsymbol{x}_t$ is also normally distributed.

Theorem 4. *Let $\boldsymbol{x}_0 \sim \mathcal{N}_p(\boldsymbol{0}, \Sigma)$ and $\boldsymbol{x}_{t+1} = \int \boldsymbol{y}p_t(\boldsymbol{y}|\boldsymbol{x}_t)d\boldsymbol{y}$. Here $p_t(\cdot)$ denotes the distribution of $\boldsymbol{x}_t$ and $p_t(\boldsymbol{y}|\boldsymbol{z}) = \phi(\boldsymbol{z} - \boldsymbol{y}; 0, \frac{h}{2}diag(1/\boldsymbol{w}^{(t)}))q_t(\boldsymbol{y})/p_t(\boldsymbol{z})$. Then, $\boldsymbol{x}_t \sim \mathcal{N}_p(\boldsymbol{0}, diag((s_1^{(t)})^2, \ldots, (s_p^{(t)})^2))$, with $s_l^{(t+1)} = (1 + h(s_l^{(t)})^2/2w_l^{(t)})^{-1}s_l^{(t)}$.*

Thus, the sequence of standard deviations $\{s_l^{(t)}\}$ form a decreasing sequence, which is bounded below. Hence, by monotone convergence theorem of real sequences (Rudin 1964), $\{s_l^{(t)}\}$ also converges. Let this limit be s_l. Hence $\{\boldsymbol{x}_l^t\}$, the l-th coordinate of $\boldsymbol{x}_t$, converges in distribution to

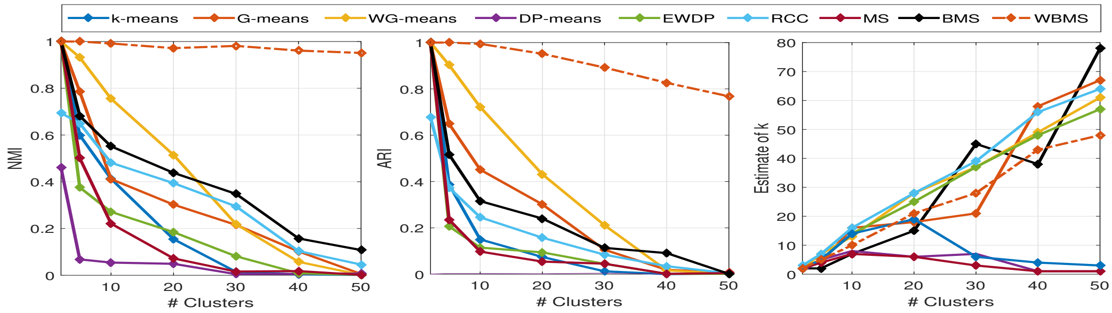

Figure 2: Performances of peer algorithms in terms of NMI (left), ARI (middle) and the estimated number of clusters (right) as the number of clusters increases in simulation 1. It is easily observed that WBMS consistently resembles the ground truth, while the other algorithms fail.

either $\mathcal{N}(0, s_l^2)$ (if $s_l > 0$) or the degenerate distribution at 0 (if $s_l = 0$). We also observe that at a population level,

$$w_l^{(t)} = \frac{\exp\{-E(x_l^{(0)} - x_l^{(t)})^2/\lambda\}}{\sum_{l'=1}^{p} \exp\{-E(x_{l'}^{(0)} - x_{l'}^{(t)})^2/\lambda\}}.$$

Since $\boldsymbol{w}_{(t)}$ is a continuous function of $\boldsymbol{v}_t$, where $\boldsymbol{v}_t = (E(x_1^{(0)} - x_1^{(t)})^2, \dots, E(x_p^{(0)} - x_p^{(t)})^2)$. Since $\boldsymbol{x}_t$ converges in distribution, $\boldsymbol{v}_t$ also converges, which in turn implies that $\boldsymbol{w}^{(t)}$ converges. Let the limit be $\boldsymbol{w}$. The following theorem asserts that $s_l^{(t)} \to 0$, as $t \to \infty$, which in turn implies that $\boldsymbol{x}_t$ converges to $\mathbf{0}$, in distribution.

Theorem 5. *Let $\boldsymbol{x}_t$, $s_l^{(t)}$ be as in Theorem 4. Then, $s_l = \lim_{t\to\infty} s_l^{(t)} = 0$ and the order of convergence of $\{s_l^{(t)}\}_{t=1}^{\infty}$ is at least cubic. Moreover, the asymptotic rate of convergence of $\{s_l^{(t)}\}$ is $\frac{2w_l}{h}$.*

Thus, the asymptotic convergence rate is smaller if w_l is smaller, meaning that if a feature is deemed to have smaller feature weight, i.e. little relevance in containing the cluster structure, the convergence along that feature is faster.

Also note that since $\boldsymbol{x}_t$ converges in distribution to $\mathbf{0}$, $w_l^{(t)}$, being a bounded continuous function of $\boldsymbol{x}_t$ also converge in distribution to

$$w_l = \frac{\exp\{-E(x_l^{(0)} - 0)^2/\lambda\}}{\sum_{l'=1}^{p} \exp\{-E(x_{l'}^{(0)} - 0)^2/\lambda\}} \propto \exp\{-\sigma_l^2/\lambda\}.$$

Thus, in limit, features with larger variances receive smaller feature weights compared to features with smaller variances.

Experimental Results

In this section, we compare WBMS with classical baselines and state-of-the-art automated clustering algorithms. To evaluate the performances, we use Normalized Mutual Information (NMI) (Vinh, Epps, and Bailey 2010) and Adjusted Rand Index (ARI) (Hubert and Arabie 1985) between the ground truth and the obtained partition. For both the indices, a value of 1 indicates perfect clustering while a value of 0 indicates completely random class labels. Since our algorithm is developed for automated clustering, apart from the baseline k-means (MacQueen 1967), we compare our method with Blurring Mean Shift (BMS) (Wang and Carreira-Perpinán 2010), Gaussian means (G-means) (Hamerly and Elkan 2004), Dirichlet Process means (DP-means) (Kulis and Jordan 2012), Entropy Weighted DP-means (EWDP) (Paul and Das 2020), and the Robust Continuous Clustering (RCC) (Shah and Koltun 2017) algorithm. For fair comparison against the other automated clustering methods, the number of clusters in k-means is supplied through the Gap statistics method (Tibshirani, Walther, and Hastie 2001). It should be noted WG-means and RCC already entails higher computational complexity and we do not consider other alternative clustering techniques, which require k as an input. Since k-means and G-means depend on the random seeding, we run each algorithm 20 times independently on each data and the report the average performance. All the datasets are normalized (z-transform) before use. In our experiments, we use the Gaussian kernel. We observed that for the proposed algorithm, $h \in (0.1, 1)$ and $\lambda \in [1, 20]$ preserves a consistently good level of performance. All the other algorithms are tuned using their standard protocols. A pertinent ablation study is provided in the supplementary document. Additional experiments and runtime comparisons also appear therein. For the sake of reproducibility, all the codes, datasets, and supplementary information are available at https://github.com/SaptarshiC98/WBMS.

Simulation Studies

We now discuss the behavior of the peer algorithms through a set of simulation studies.

Simulation 1: Effect of increasing k We now examine the behavior of the WBMS algorithm as the number of cluster k increases. In this simulation study, we take $n = 20 \times k$, $p = 20$, while k varies from 2 to 50. Let $\boldsymbol{\Theta}$ be the $k \times p$ real matrix, whose rows represent the k cluster centroids. We generate $\boldsymbol{\Theta}$ as follows.

- Generate $\theta_{jl} \sim Unif(0, 1)$ independently for $j = 1, \dots, k$ and $l = 1, \dots, 5$.
- Set $\theta_{jl} = 0$ for $j = 1, \dots, k$ and $l = 6, \dots, 20$.

After generating $\boldsymbol{\Theta}$, we generate the $n \times p$ data matrix $\boldsymbol{X}$ as:

$$c_i \sim Unif(\{1, \dots, k\}); x_{il} \sim \mathcal{N}_1(\theta_{c_i,l}, 0.02^2)$$

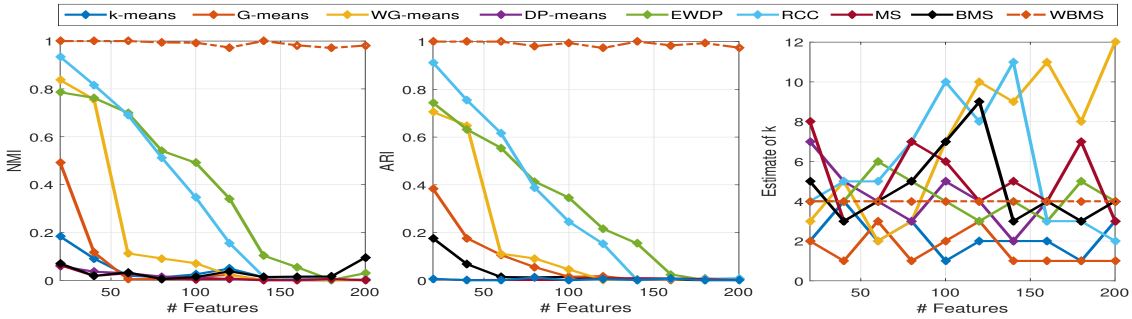

Figure 3: Performances of peer algorithms in terms of NMI (left), ARI (middle) and the estimated number of clusters (right) as the number of features increases in simulation 2. It is easily observed that WBMS consistently resembles the ground truth, while the other algorithms fail as the dimensionality of the data increases.

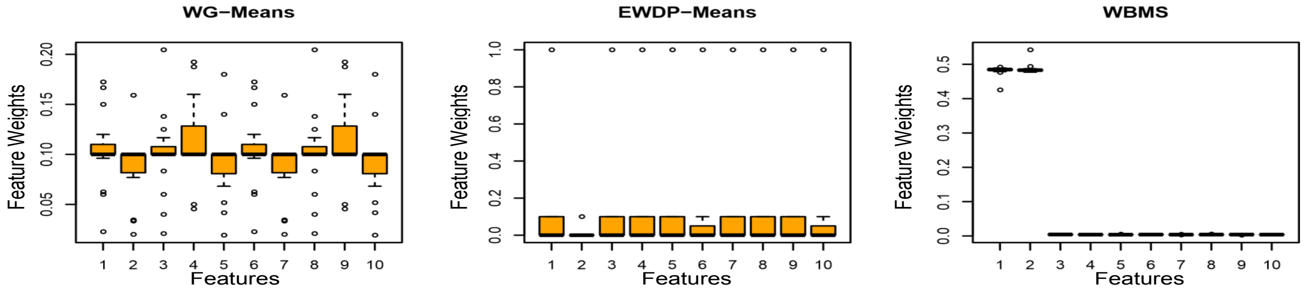

Figure 4: Boxplot shows that WBMS consistently identifies true features while WG-means and EWDP-means fail to do so.

if $l \in \{1, \dots, 5\}$; $x_{il} \sim \mathcal{N}_1(0, 1)$ if $l \in \{6, \dots, 20\}$.

From the data generation procedure, it is easy to observe that only the first five features contain the cluster structure of the data. We run all the peer algorithms on each of the datasets. Fig. 2 shows the average NMI and ARI values between the ground truth and the obtained partition. We also plot the average estimated number of clusters ($\hat{k}$) against the true number of clusters (k). It can be easily observed that WBMS consistently outperforms the other peer algorithms not only in terms of clustering performance but also in finding the true number of clusters.

Simulation 2: Effect of increasing *p* This experiment assesses the performance of WBMS as the number of features grows. We generate $n = 100$ observations with $k = 4$ clusters. We increase the dimension from $p = 20$ to $p = 200$ at differences of 20 to observe the effect of growing features. In each case, the number of informative features is fixed at 5% of p with an exception for $p = 20$, where we take 2 informative features. The clusters are spherical without any overlaps and are generated from Normal distributions with variance 0.3 and means generated from $Unif(0, 1)$ distribution. The non-informative features are generated from $\mathcal{N}_1(0, 1)$. Fig. 3 compares the NMI, ARI and estimated number of clusters for all the peer algorithms. From Fig. 3, we can easily observe that among all peer algorithms, not only the WBMS algorithm performs best in terms of NMI and ARI values, it also estimates the number of clusters perfectly.

Simulation 3: Feature Selection We now examine the feature weighting properties of WBMS. Here, we take $n = 100$, $p = 10$, $k = 4$ and follow the data generation procedure described in Simulation 2. We take the first two features to be informative that actually contain the cluster structure. We compare the feature weights obtained by WBMS against those obtained by WG-means and EWDP. We record the feature weights obtained by these two algorithms along with proposed WBMS over 100 replicates of the data. The box plots for these 100 optimal feature weights are shown in Fig. 4 for all the three algorithms. The proposed WBMS successfully assigns almost all weights to informative features 1 and 2, even in this low signal-to-noise-ratio situation. Meanwhile, its peers WG-means and EWDP fail to do so.

Case Study on Glioma Data

We now evaluate the performance of our algorithm with a specific case study on the microarray dataset, Glioma (Nutt et al. 2003). The dataset comprises of 4434 gene expression levels, collected over 50 samples. There are four natural classes in the data viz. Glioblastomas (CG), Non-cancer Glioblastomas (NG), Cancer Oligodendrogliomas (CO), and Non-cancer Oligodendrogliomas (NO). In our experimental study, we compare our proposed WBMS algorithm to all the algorithms described in beginning of this section. To visualize the clustering results, we perform a *t-SNE* (Maaten and Hinton 2008) and show the resulting embedding in Fig. 5,

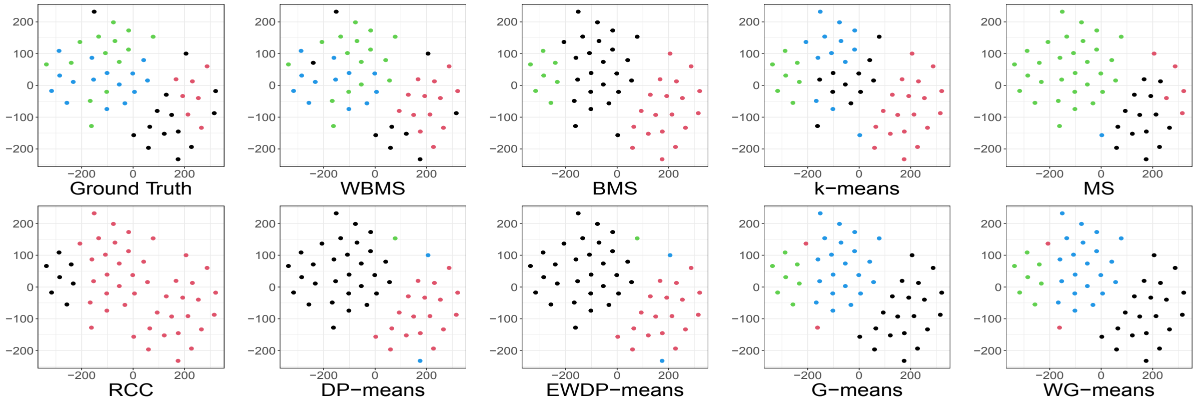

Figure 5: *t-SNE* plots for the GLIOMA dataset, color-coded by the partitions obtained by each peer algorithm.

Datasets	Method	k-Means	G-Means	WG-Means	DP-Means	EWDP	RCC	MS	BMS	WBMS
GLIOMA	NMI	0.499	0.522	0.517	0.576	0.675	0.113	0.580	0.546	**0.706**
	ARI	0.328	0.367	0.373	0.416	0.598	0.004	0.429	0.398	**0.618**
Appendicitis	NMI	0.157	0.165	0.185	0.158	0.189	0.193	0.008	0.195	**0.249**
	ARI	0.230	0.169	0.188	0.231	0.188	0.000	0.014	0.223	**0.434**
Zoo	NMI	0.741	0.801	0.749	0.611	0.859	0.557	0.706	0.841	**0.925**
	ARI	0.459	0.646	0.559	0.452	0.872	0.039	0.436	0.867	**0.953**
Mammographic	NMI	0.231	0.181	0.240	0.191	0.273	0.181	0.181	0.008	**0.348**
	ARI	0.292	0.209	0.293	0.257	0.247	0.001	0.150	0.001	**0.351**
Yale	NMI	0.539	0.417	0.521	0.324	0.334	0.268	0.548	0.222	**0.693**
	ARI	0.271	0.173	0.227	0.040	0.061	0.031	0.069	0.027	**0.485**
nci9	NMI	0.458	0.394	0.394	0.181	0.211	0.143	0.346	0.394	**0.686**
	ARI	0.187	0.100	0.100	0.005	0.018	0.005	0.011	0.086	**0.419**
Lymphoma	NMI	0.441	0.592	0.690	0.518	0.648	0.243	0.241	0.595	**0.778**
	ARI	0.269	0.340	0.463	0.088	0.431	0.001	0.000	0.458	**0.604**
Movement Libras	NMI	0.591	0.231	0.328	0.333	0.465	0.639	0.245	0.503	**0.663**
	ARI	0.309	0.068	0.123	0.113	0.217	0.014	0.090	0.185	**0.532**
GCM	NMI	0.532	0.484	0.497	0.025	0.497	0.637	0.456	0.649	**0.833**
	ARI	0.288	0.248	0.266	0.002	0.266	0.361	0.413	0.536	**0.714**

Table 1: Performance Analysis on Real Life Datasets in terms of NMI & ARI values.

color-coded with the partitions obtained from the peer algorithms. WBMS closely represents the ground truth compared to its competitors. This is also be seen from Table 1.

Performance on Real Data Benchmarks

To further demonstrate the efficacy of our proposal, we compare WBMS with the peer algorithms on eight benchmark real datasets. The datasets are taken from the UCI machine learning repository (Dua and Graff 2017) , Keel repository (Alcalá-Fdez et al. 2011)^1 and ASU feature selection repository (Li et al. 2018). The microarray GCM data is collected from (Ramaswamy et al. 2001). The data dimensions are reported in the supplement. We follow the same computational protocols as stated at the beginning of Section . The performance of each algorithm, in terms of NMI and ARI, is reported in Table 1, which clearly indicate the superior performance of WBMS on each of the nine real data benchmarks.

Discussions

Despite decades of advancement, there is no proper solution for finding the number of clusters and feature weights simultaneously in the clustering problem. For high dimensional datasets, which contain a significant number of non-informative features, most of the existing automated clustering methods fail to identify the discriminating features as well as the number of clusters, resulting in poor performance. To circumvent such difficulties, in this paper, we put forth a novel clustering algorithm known as the Weighted Blurring Mean Shift (WBMS), which not only provides an efficient feature representation of the data but also detects the number of clusters simultaneously. WBMS comes with closed-form updates and rigorous convergence guarantees. We have also mathematically proved that under the nominal assumption of normality of the clusters, WBMS has at least a cubic convergence rate. Through detailed experimental analysis on simulated and real data, WBMS is particularly shown to be useful for high-dimesnional data with many clusters.

Ethics Statement

To the best of our knowledge, the authors abide by the AAAI Publications Ethics and Malpractice Statement and the AAAI Code of Professional Conduct. This paper focuses on algorithmic and theoretical contributions to unsupervised learning of high-dimensional datasets in an automated manner. There are no immediate privacy or ethical concerns, but by addressing a well-known problem of finding the number of clusters and informative features simultaneously in the paradigm of unsupervised learning, broader impacts extend beyond methodological contributions and may have wider impacts on the learning community.

Clustering problems have been addressed for decades and still, in this era of constantly growing data dimensions, there are only a few contributions in finding the number of clusters and informative features simultaneously. Clustering has been used for countless applications, including community detection, drug discovery, and gene identification for cancers and other diseases. In such settings where the interpretations and decisions based on clustering solutions have a significant scientific and societal bearing, unknown datasets must identify the groups correctly without any previous knowledge whatsoever. By ameliorating the widely popular blurring mean shift clustering methods which provide equal importance to all the features, our contributions, in turn, mitigate the 'curse of dimensionality' by providing appropriate feature weights to the relevant attributes. Being said that, it is always important for the practitioner to look deeply into the results produced by our or any other statistical learning algorithm and should consider all unforeseen repercussions before jumping to any conclusion.

References

Alcalá-Fdez, J.; Fernández, A.; Luengo, J.; Derrac, J.; García, S.; Sánchez, L.; and Herrera, F. 2011. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework[1]. *Journal of Multiple-Valued Logic & Soft Computing* 17.

Archambeau, C.; and Verleysen, M. 2007. Robust bayesian clustering. *Neural Networks* 20(1): 129–138.

Carlsson, G.; and Mémoli, F. 2010. Characterization, Stability and Convergence of Hierarchical Clustering Methods. *J. Mach. Learn. Res.* 11: 1425–1470. ISSN 1532-4435.

Carreira-Perpiñán, M. Á. 2006. Fast nonparametric clustering with Gaussian blurring mean-shift. In *Proceedings of the 23rd international conference on Machine learning*, 153–160.

Chakraborty, S.; and Das, S. 2018. Simultaneous variable weighting and determining the number of clusters—A weighted Gaussian means algorithm. *Statistics & Probability Letters* 137: 148–156.

Chakraborty, S.; Paul, D.; Das, S.; and Xu, J. 2020. Entropy weighted power k-means clustering. In *International Conference on Artificial Intelligence and Statistics*, 691–701. PMLR.

Chen, T.-L. 2015. On the convergence and consistency of the blurring mean-shift process. *Annals of the Institute of Statistical Mathematics* 67(1): 157–176.

Cheng, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence* 17(8): 790–799.

Chi, E. C.; Allen, G. I.; and Baraniuk, R. G. 2017. Convex biclustering. *Biometrics* 73(1): 10–19.

Chi, E. C.; and Lange, K. 2015. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics* 24(4): 994–1013.

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

Donoho, D. L.; et al. 2000. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture* 1(2000): 32.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository[1] . URL http://archive.ics.uci.edu/ml.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(11): 2765–2781.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, 226–231.

Fischer, A. 2011. On the number of groups in clustering. *Statistics & Probability Letters* 81(12): 1771–1781.

Gupta, A.; Datta, S.; and Das, S. 2018. Fast automatic estimation of the number of clusters from the minimum intercenter distance for k-means clustering. *Pattern Recognition Letters* 116: 72 – 79. ISSN 0167-8655.

Hamerly, G.; and Elkan, C. 2004. Learning the k in k-means. In *Advances in neural information processing systems*, 281–288.

Hess, S.; Duivesteijn, W.; Honysz, P.; and Morik, K. 2019. The SpectACl of nonconvex clustering: a spectral approach to density-based clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3788–3795.

Huang, J. Z.; Ng, M. K.; Rong, H.; and Li, Z. 2005. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(5): 657–668.

Huang, K.; Fu, X.; and Sidiropoulos, N. D. 2018. On Convergence of Epanechnikov Mean Shift. In *AAAI*.

Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of classification* 2(1): 193–218.

Jain, A. K. 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31(8): 651 – 666. ISSN 0167-8655. doi:https://doi.org/10.1016/j.patrec.2009.

[1] Last accessed on Sep 09, 2020

09.011. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

Jiang, H. 2017. Density level set estimation on manifolds with dbscan. *arXiv preprint arXiv:1703.03503* .

Jin, J.; Wang, W.; et al. 2016. Influential features PCA for high dimensional clustering. *The Annals of Statistics* 44(6): 2323–2359.

Kriegel, H.-P.; Kröger, P.; and Zimek, A. 2009. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 3(1): 1–58.

Kulis, B.; and Jordan, M. I. 2012. Revisiting k-means: New Algorithms via Bayesian Nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. URL http://icml.cc/2012/papers/291.pdf.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2018. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* 50(6): 94.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

McNicholas, P. D. 2016. Model-based clustering. *Journal of Classification* 33(3): 331–373.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 849–856.

Nutt, C. L.; Mani, D.; Betensky, R. A.; Tamayo, P.; Cairncross, J. G.; Ladd, C.; Pohl, U.; Hartmann, C.; McLaughlin, M. E.; Batchelor, T. T.; et al. 2003. Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer research* 63(7): 1602–1607.

Paul, D.; and Das, S. 2020. A Bayesian Non-parametric Approach for Automatic Clustering with Feature Weighting. *Stat* e306.

Pelckmans, K.; De Brabanter, J.; Suykens, J. A.; and De Moor, B. 2005. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*.

Ramaswamy, S.; Tamayo, P.; Rifkin, R.; Mukherjee, S.; Yeang, C.-H.; Angelo, M.; Ladd, C.; Reich, M.; Latulippe, E.; Mesirov, J. P.; et al. 2001. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences* 98(26): 15149–15154.

Rocha, P.; Pinheiro, D.; Cadeiras, M.; and Bastos-Filho, C. 2020. Towards Automatic Clustering Analysis using Traces of Information Gain: The InfoGuide Method. *arXiv preprint arXiv:2001.08677* .

Rudin, W. 1964. *Principles of mathematical analysis*, volume 3. McGraw-hill New York.

Sarkar, S.; and Ghosh, A. K. 2019. On perfect clustering of high dimension, low sample size data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Shah, S. A.; and Koltun, V. 2017. Robust continuous clustering. *Proceedings of the National Academy of Sciences* 114(37): 9814–9819.

Su, M.; and Shang, Y. 2017. Solving fixed-point problems with inequality and equality constraints via a non-interior point homotopy path-following method. *Mathematical Problems in Engineering* 2017.

Tibshirani, R.; Walther, G.; and Hastie, T. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63(2): 411–423.

Vedaldi, A.; and Soatto, S. 2008. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, 705–718. Springer.

Vinh, N. X.; Epps, J.; and Bailey, J. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research* 11: 2837–2854.

Wang, B.; Zhang, Y.; Sun, W. W.; and Fang, Y. 2018. Sparse convex clustering. *Journal of Computational and Graphical Statistics* 27(2): 393–403.

Wang, W.; and Carreira-Perpinán, M. A. 2010. Manifold blurring mean shift algorithms for manifold denoising. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1759–1766.

Witten, D. M.; and Tibshirani, R. 2010. A framework for feature selection in clustering. *Journal of the American Statistical Association* 105(490): 713–726.

Xu, D.; and Tian, Y. 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science* 2(2): 165–193.

Xu, J.; and Lange, K. 2019. Power k-means clustering. In *International Conference on Machine Learning*, 6921–6931.