On Convergence of Epanechnikov Mean Shift

Kejun Huang
University of Minnesota
Minneapolis, MN 55414
huang663@umn.edu

Xiao Fu
Oregon State University
Corvallis, OR 97331
xiao.fu@oregonstate.edu

Nicholas D. Sidiropoulos
University of Virginia
Charlottesville, VA 22904
nikos@virginia.edu

Abstract

Epanechnikov Mean Shift is a simple yet empirically very effective algorithm for clustering. It localizes the centroids of data clusters via estimating modes of the probability distribution that generates the data points, using the 'optimal' Epanechnikov kernel density estimator. However, since the procedure involves *non-smooth* kernel density functions, the convergence behavior of Epanechnikov mean shift lacks theoretical support as of this writing—most of the existing analyses are based on smooth functions and thus cannot be applied to Epanechnikov Mean Shift. In this work, we first show that the original Epanechnikov Mean Shift may indeed terminate at a non-critical point, due to the non-smoothness nature. Based on our analysis, we propose a simple remedy to fix it. The modified Epanechnikov Mean Shift is guaranteed to terminate at a local maximum of the estimated density, which corresponds to a cluster centroid, within a *finite* number of iterations. We also propose a way to avoid running the Mean Shift iterates from every data point, while maintaining good clustering accuracies under non-overlapping spherical Gaussian mixture models. This further pushes Epanechnikov Mean Shift to handle very large and high-dimensional data sets. Experiments show surprisingly good performance compared to the Lloyd's K-means algorithm and the EM algorithm.

Introduction

Clustering is a fundamental problem in artificial intelligence and statistics (Jain, Murty, and Flynn 1999). The simplest form is arguably the K-means clustering, in which a set of data points $\{\boldsymbol{x}_m\}_{m=1}^M \subseteq \mathbb{R}^d$ is given, and the objective is to separate them into K clusters, such that the sum of the cluster variances is minimized. It has been shown that K-means clustering is NP-hard in general (Aloise et al. 2009; Dasgupta and Freund 2009), even though Lloyd's algorithm usually gives reasonably good approximate solutions (Lloyd 1982) when d is small. In fact, it has been used as a standard sub-routine for more complicated clustering tasks like spectral clustering (Ng, Jordan, and Weiss 2001) and subspace clustering (Elhamifar and Vidal 2013), despite the fact that it is not guaranteed to give the global optimal solution.

Several attempts have been made to quantify cases under which we can provably cluster the data under a probabilistic generative model. Based on the Gaussian mixture model (GMM), instead of applying Lloyd's algorithm or Expectation-Maximization (Dempster, Laird, and Rubin 1977), this line of work devises sophisticated and somewhat conceptual methods that guarantee correct estimation of the GMM parameters under additional conditions. The first of this genre, to the best of our knowledge, is the work in (Dasgupta 1999), which shows that if the Gaussian components are almost disjoint, then the deflation-type method in (Dasgupta 1999) correctly clusters the data with high probability. A more practical approach is later proposed in (Dasgupta and Schulman 2000) with the same performance guarantee, which is a two-round variant of the EM algorithm. A number of follow-up works try to further improve the bound on how close the Gaussian components can be (Arora and Kannan 2001; Vempala and Wang 2004; Brubaker and Vempala 2008; Moitra and Valianty 2010; Belkin and Sinha 2010), but most of them focus on theoretical guarantees but not practical implementations.

On the other hand, there is a simple and effective method for clustering called Mean Shift (Fukunaga and Hostetler 1975; Cheng 1995; Comaniciu and Meer 2002) that has been popular in the field of computer vision. Two main versions of Mean Shift are the Epanechnikov Mean Shift and Gaussian Mean Shift, and the details will be explained in the next section. Compared to K-means and Gaussian mixture model, fewer theoretical results have been presented regarding Mean Shift. Compared to Gaussian Mean Shift, even less analysis exists for Epanechnikov Mean Shift, partly due to its non-smoothness nature, despite its simplicity and effectiveness.

The main contribution of this paper is the establishment of convergence for Epanechnikov Mean Shift. There have been many convergence studies on different variants and approximations to the original Epanechnikov Mean Shift. However, to the best of our knowledge, there is no analysis that directly addresses the original Epanechnikov Mean Shift, partially because non-smoothness of the kernel employed in the method poses a hard analytical problem. Nevertheless, analyzing the convergence behavior of the original Epanechnikov Mean Shift is of great interest since it is based on the 'optimal' kernel in density estimation. In this work, we provide detailed functional analysis and rigorous proof for Epanechnikov Mean Shift. We first show that the method in-

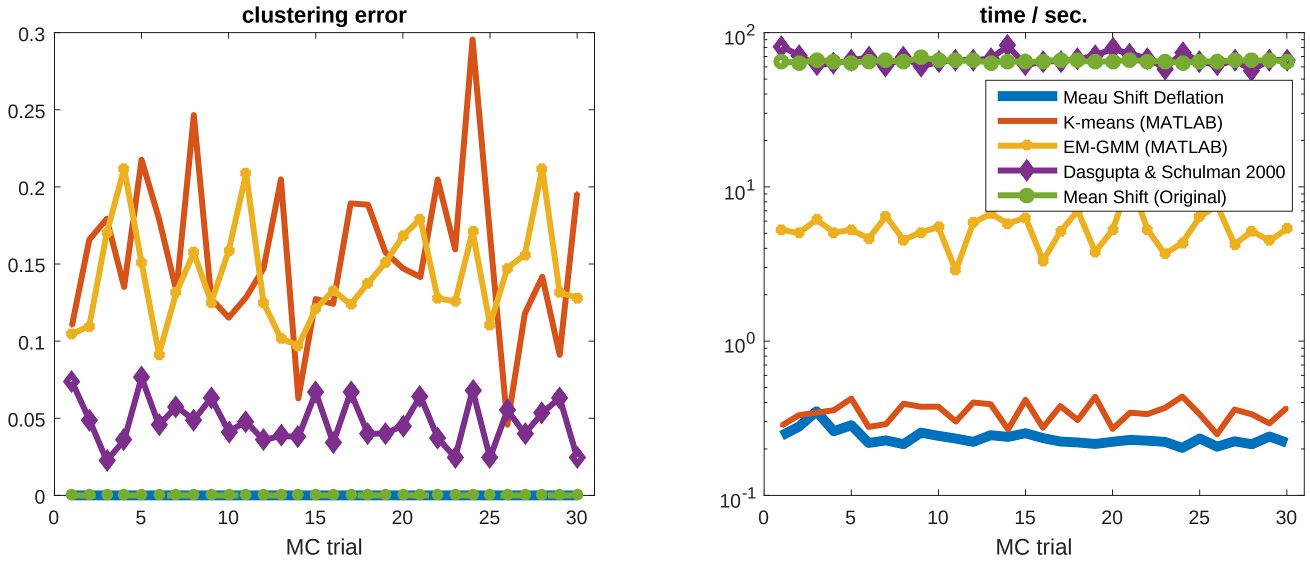

Figure 1: 100 Monte-Carlo simulations on synthetic data. Left: clustering error; right: run time.

deed can get swamped at some undesired points, but with very simple modification it is guaranteed to reach a local optimum of the data density function that corresponds to a cluster centroids *within finite number of iterations*. This is the first analytical result that backs convergence behaviors of the Epanechnikov Mean Shift and the proposed modification is of practical significance. Inspired by the idea of deflation by (Dasgupta 1999) and follow-up works, we also propose a deflation-variant of the Mean Shift algorithm that is guaranteed to correctly cluster the data one group at a time under some additional conditions. This strategy saves a huge amount of computations compared to the original version and thus suits large-scale clustering problems.

Illustrative Example

Before we delve into convergence analysis of Epanechnikov Mean Shift, we give a simple illustrative example to showcase its effectiveness in clustering—which explains the reason why this particular method interests us. Specifically, we test the performance of the proposed deflation-based Epanechnikov Mean Shift and some classic clustering methods, including Lloyd's K-means algorithm, Expectation-Maximization (EM) for Gaussian mixture models (GMM), the two-round variant of EM by (Dasgupta and Schulman 2000), and the original Epanechnikov Mean Shift. The experiment is conducted in MATLAB, with the build-in implementation of K-means clustering and EM for GMM. Notice that these are well-implemented K-means / EM algorithms, with smart initialization suggested by K-means++ (Arthur and Vassilvitskii 2007), and/or various parallel implementation / multiple re-start enhancements that have been shown to work well in practice. The two-round variant of EM (Dasgupta and Schulman 2000) is mathematically proven to work well with high probability when the clusters are non-overlapping.

The experiments are conducted on a synthetic dataset $\{\boldsymbol{x}_m\}_{m=1}^{M} \subseteq \mathbb{R}^d$ generated as follows. For $d = 100$, we prescribe $K = 30$ clusters (Gaussian components). For cluster k, we first randomly generate its centroid $\boldsymbol{\mu}_k \sim \mathcal{N}(0, 4\boldsymbol{I})$, and then generate $M_k = 50k$ i.i.d. data points from $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{I})$. Then we hide the cluster labels and feed the data into various algorithms. For the two versions of the Mean Shift algorithm, there is no need to indicate the number of clusters K before hand—the only parameter (kernel bandwidth) is tuned by leave-one-out cross-validation, and then the algorithm automatically detects the number of clusters in the data set. For the other methods, the correct number of clusters K is given, which means they are already using more information than the Mean Shift-based methods. The procedure is repeated 30 times. In each Monte-Carlo trial, the obtained cluster labels are aligned with the true labels using the Hungarian algorithm (Kuhn 1955). The clustering error is then calculated as the ratio of wrongly labeled data points over the total number of data points.

The clustering error and runtime for each Monte-Carlo trial are shown in Figure 1. The first observation is that plain vanilla K-means and EM do not cluster the data very accurately, even though there exists a good clustering structure according to how we generate the data. The clustering error is greatly reduced if we adopt the two-round variant of EM (Dasgupta and Schulman 2000), with the compromise of a significantly higher amount of computation time. However, since the different Gaussian components are just marginally separated (and the sizes of each cluster are somewhat unbalanced), the performance is not as good as we might expect according to (Dasgupta and Schulman 2000). Mean Shift-based methods, on the other hand, give the surprising zero clustering error in all cases; considering the fact that the correct number of clusters K is not given to these methods, the results look even more impressive. In terms of computation time, the original Mean Shift takes similar time as that of (Dasgupta and Schulman 2000), whereas the proposed Mean Shift deflation takes, remarkably, *the least* amount of time, even compared to the simple K-means.

In this paper, we will study convergence properties of the original Epanechnikov Mean Shift and its deflation variant, and explain the reasons behind its effectiveness.

Background: KDE and Mean Shift

The intuition behind Mean Shift for clustering is as follows. Suppose we have the probability density function (PDF) $p(\boldsymbol{z})$ of the dataset $\{\boldsymbol{x}_m\}_{m=1}^M \subseteq \mathbb{R}^d$. If the PDF $p(\boldsymbol{z})$ has K modes, then we expect K clusters in this dataset. Furthermore, if we run an optimization algorithm, e.g. gradient descent, initialized at a data point $\boldsymbol{x}_m$, and it converges to the k-th mode, then we declare that $\boldsymbol{x}_m$ belongs to the k-th cluster.

Kernel Density Estimation

In practice, we do not have access to the PDF $p(\boldsymbol{z})$, but only the set of data points $\{\boldsymbol{x}_m\}_{m=1}^M$. To implement the aforementioned intuition, one needs to first estimate the PDF $p(\boldsymbol{z})$—this is called density estimation (Scott 2015). The most popular approach for density estimation is the so-called kernel density estimator (KDE). For a given kernel function $K(\boldsymbol{z})$ that satisfies

$$K(\boldsymbol{z}) \geq 0 \text{ and } \int K(\boldsymbol{z}) d\boldsymbol{z} = 1,$$

the corresponding KDE is simply

$$\widehat{p}(\boldsymbol{z}) = \frac{1}{M} \sum_{m=1}^{M} K(\boldsymbol{z} - \boldsymbol{x}_m).$$

Two popular choices of the kernels are the Gaussian kernel

$$K_G(\boldsymbol{z}; w) = \frac{c}{w^d} \exp\left(-\frac{\|\boldsymbol{z}\|^2}{2w^2}\right) \tag{1}$$

and the Epanechnikov kernel

$$K_E(\boldsymbol{z}; w) = \frac{c}{w^d} \left[1 - \frac{\|\boldsymbol{z}\|^2}{w^2}\right]_+, \tag{2}$$

where c in (1) and (2) are normalizing constants ensuring that the kernel integrates to one. Each of them (and all other kernels) are parameterized by a scalar w, called the *bandwidth*, which controls the variance of the kernel. It has been shown that the Epanechnikov kernel asymptotically minimizes the mean squared error (MSE)

$$\int \left(p(\boldsymbol{z}) - \widehat{p}(\boldsymbol{z})\right)^2 d\boldsymbol{z}, \tag{3}$$

among all possible kernel functions (Epanechnikov 1969). Somewhat surprisingly, this 'optimal' kernel is a highly non-smooth function.

The bandwidth of the kernel w plays an important role on how well the KDE approximates the true density. In practice, one can adopt the *leave-one-out cross validation* approach to determine this parameter, as we did in this work. The MSE of the estimated density (and the unknown true density) (3) can be separated into three terms:

$$\int p^2(\boldsymbol{z}) d\boldsymbol{z} + \int \widehat{p}^2(\boldsymbol{z}) d\boldsymbol{z} - 2\mathbb{E}\{\widehat{p}(\boldsymbol{z})\}.$$

The first term is unknown, but a constant; the second term can be directly calculated; and the third term is estimated via leave-one-out cross-validation. This quantity is evaluated at a set of values for w, and the one that gives the minimum value is selected as the bandwidth for the KDE.

Mean Shift

Based on the KDE $\widehat{p}(\boldsymbol{z})$, the Mean Shift algorithm tries to find modes of $\widehat{p}(\boldsymbol{z})$ via the following (weighted average) iterates (Fukunaga and Hostetler 1975; Cheng 1995; Comaniciu and Meer 2002), initialized at each $\boldsymbol{x}_m$:

$$\boldsymbol{z} \leftarrow \frac{1}{\sum_{m=1}^{M} g(\|\boldsymbol{z} - \boldsymbol{x}_m\|^2)} \sum_{m=1}^{M} g(\|\boldsymbol{z} - \boldsymbol{x}_m\|^2)\boldsymbol{x}_m,$$

where $g(\cdot)$ is called the profile for the kernel function $K(\cdot)$. Putting details aside, the profile for the Gaussian kernel is $\exp(\|\boldsymbol{z} - \boldsymbol{x}_m\|^2/2w^2)$, and that for the Epanechnikov kernel is the indicator function $\mathbb{1}(\|\boldsymbol{z} - \boldsymbol{x}_m\|^2 < w^2)$.

Existing analyses for convergence properties of Mean Shift are mostly based on smooth optimization, and argue that the update is always going at the gradient ascent direction. Borrowing the convergence results for gradient-based methods, it is then claimed that the Mean Shift iterates converges to a local maximum of $\widehat{p}(\boldsymbol{z})$. On hindsight, we make the following comments:

1. The *optimal* Epanechnikov kernel is non-smooth, so the existing convergence claims cannot establish convergence to a local optimum—which asymptotically approaches a mode of $p(\boldsymbol{z})$. In fact, we will show that the plain vanilla Epanechnikov Mean Shift may indeed get stuck at a non-critical point, and we will provide a simple remedy to fix it.
2. For smooth kernels like the Gaussian kernel, it is indeed easy to show that the algorithm converges to a stationary point. However, not all stationary points are local optima—there may exist saddle points, and there is in general no simple way to check whether it is a local optimum or not.
3. An interesting observation is that Mean Shift with smooth kernels usually converges slower than Epanechnikov kernel. The convergence rate for Gaussian Mean Shift can be as slow as sub-linear (Carreira-Perpiñán 2007), whereas Epanechnikov Mean Shift *terminates* in finite number of steps (Comaniciu and Meer 2002), although a rigorous proof for this claim is still missing.

The remainder of the paper tries to bridge the gap between the good empirical performance and lack of rigorous theoretical analysis for the Epanechnikov Mean Shift. We show that, with a simple modification, Epanechnikov Mean Shift

Algorithm 1 Epanechnikov Mean Shift

```
Require: ${\boldsymbol{x}_m}_{m=1}^M$, $w^2$
1: for $m = 1, ..., M$ do
2:     initialize $\boldsymbol{z}_m \leftarrow \boldsymbol{x}_m$
3:     repeat                    ▷ Epanechnikov Mean Shift iterates
4:         $\mathcal{I}(\boldsymbol{z}_m) \leftarrow \{i \in [M] : \|\boldsymbol{x}_i - \boldsymbol{z}_m\|^2 < w^2\}$
5:         $\boldsymbol{z}_m \leftarrow \frac{1}{|\mathcal{I}(\boldsymbol{z}_m)|} \sum_{i \in \mathcal{I}(\boldsymbol{z}_m)} \boldsymbol{x}_i$
6:     until convergence (cf. Alg. 2)
7: end for
8: find $K$ distinct vectors in $\{\boldsymbol{z}_m\}_{m=1}^M$, denote as $\{\boldsymbol{\mu}_k\}_{k=1}^K$
9: $\boldsymbol{x}_m$ in cluster $k$ if $\boldsymbol{z}_m = \boldsymbol{\mu}_k$.
```

terminates at a local maximum of $\widehat{p}(\boldsymbol{z})$ within finite number of iterations. Even though the objective function is nonconvex and non-smooth, the convergence result is surprisingly strong: It is guaranteed to terminate at a local optimum, *never* at a saddle point, and the number of iterations is finite, with *zero* precision accuracies.

For completeness, the Epanechnikov Mean Shift is clearly written in Algorithm 1. We shall call the iterative procedure between line 3–6 "Epanechnikov Mean Shift *iterates*", and the entire algorithm as Epanechnikov Mean Shift, which initializes the iterates at every data point $\boldsymbol{x}_m$.

Function Analysis

The Epanechnikov Mean Shift iterates tries to find modes (i.e., local maxima) of the KDE $\widehat{p}(\boldsymbol{z}) = \sum K_E(\boldsymbol{z} - \boldsymbol{x}_m; w)$ with the Epanechnikov kernel. As per conventions in the field of optimization, we define functions ϕ and f by flipping the sign of K_E and $\widehat{p}$, and omitting constants and scalings, which do not affect the task of optimization:

$$\phi(\boldsymbol{z}) = \min(\|\boldsymbol{z}\|^2, w^2), \tag{4}$$

$$f(\boldsymbol{z}) = \sum_{m=1}^{M} \phi(\boldsymbol{x}_m - \boldsymbol{z}). \tag{5}$$

Obviously, modes of $\widehat{p}(\boldsymbol{z})$ correspond to local minima of $f(\boldsymbol{z})$. We start by analyzing the basic properties of the loss function (5).

Lemma 1. *The function $f(\boldsymbol{z})$ is smooth almost everywhere.*

Proof. Notice that $f(\boldsymbol{z})$ is a summation of component functions $\phi(\boldsymbol{x}_m - \boldsymbol{z})$, so $f(\boldsymbol{z})$ is smooth at $\boldsymbol{z}$ if and only if $\forall\ m = 1, ..., M$, $\phi(\boldsymbol{x}_m - \boldsymbol{z})$ is smooth at $\boldsymbol{z}$. According to the definition of ϕ in (4), $\phi(\boldsymbol{x}_m - \boldsymbol{z})$ is non-smooth iff $\|\boldsymbol{x}_m - \boldsymbol{z}\|^2 = r$, which forms a set that has Lebesgue measure zero in $\mathbb{R}^d$. Because $\{\boldsymbol{x}_m\}_{m=1}^M$ is a finite set, the union set

$$\{\boldsymbol{z} : \|\boldsymbol{x}_1 - \boldsymbol{z}\|^2 = w^2\} \cup ... \cup \{\boldsymbol{z} : \|\boldsymbol{x}_M - \boldsymbol{z}\|^2 = w^2\},$$

which forms the set of non-smooth points for $f(\boldsymbol{z})$, also has Lebesgue measure zero. In other words, the function $f(\boldsymbol{z})$ is smooth almost everywhere. □

Lemma 2. *At every smooth point $\boldsymbol{z}$ of $f(\boldsymbol{z})$, define $\mathcal{I}(\boldsymbol{z}) = \{i : \|\boldsymbol{x}_i - \boldsymbol{z}\|^2 < w^2\}$, then we have*

$$\nabla f(\boldsymbol{z}) = \sum_{i \in \mathcal{I}(\boldsymbol{z})} 2(\boldsymbol{z} - \boldsymbol{x}_i), \tag{6}$$

$$\nabla^2 f(\boldsymbol{z}) = 2|\mathcal{I}(\boldsymbol{z})|\boldsymbol{I}. \tag{7}$$

Therefore, $f(\boldsymbol{z})$ is locally convex at every smooth point, and strongly convex if $\mathcal{I}$ is not empty.

Proof. If $\boldsymbol{z}$ is a smooth point, there does not exist a $\boldsymbol{x}_j$ such that $\|\boldsymbol{x}_j - \boldsymbol{z}\|^2 = w^2$. The expressions for the gradient and Hessian are elementary. For a small hyper-ball containing only smooth points, the index set $\mathcal{I}(\boldsymbol{z})$ remains the same in this convex region, therefore the Hessian remains the same in this area. Since $\nabla^2 f(\boldsymbol{z}) \succeq 0$, the function $f(\boldsymbol{z})$ is locally convex. Furthermore, if $\mathcal{I}(\boldsymbol{z}) \neq \emptyset$, then $\nabla^2 f(\boldsymbol{z}) \succeq \boldsymbol{I}$, in which case $f(\boldsymbol{z})$ is locally strongly convex. □

We now switch our focus to the non-smooth points of $f(\boldsymbol{z})$. To study their properties, we use the concept of *directional derivative*, which is defined as (Bertsekas 1999)

$$f'(\boldsymbol{z}; \boldsymbol{\delta}) \triangleq \lim_{\alpha \downarrow 0} \frac{f(\boldsymbol{z} + \alpha\boldsymbol{\delta}) - f(\boldsymbol{z})}{\alpha}, \tag{8}$$

for a particular direction $\boldsymbol{\delta}$, if the limit exists. The definition (8) clearly shows that $\boldsymbol{\delta}$ is a descending direction if $f'(\boldsymbol{z}; \boldsymbol{\delta}) < 0$. The directional derivative obeys the sum rule

$$f'(\boldsymbol{z}; \boldsymbol{\delta}) = \sum_{m=1}^{M} \phi'(\boldsymbol{x}_m - \boldsymbol{z}; \boldsymbol{\delta}). \tag{9}$$

Furthermore, if f is smooth at a point $\boldsymbol{z}$, then the directional derivative is simply $f'(\boldsymbol{z}; \boldsymbol{\delta}) = \nabla f(\boldsymbol{z})^\top \boldsymbol{\delta}$. For a non-smooth function, we can define a stationary point as follows (Razaviyayn, Hong, and Luo 2013):

Definition 1. The point $\boldsymbol{z}$ is a stationary point of $f(\cdot)$ if $f'(\boldsymbol{z}; \boldsymbol{\delta}) \geq 0$ for all $\boldsymbol{\delta}$.

Notice that if $\boldsymbol{z}$ is a smooth point for f, Definition 1 reduces to $\nabla f(\boldsymbol{z})^\top \boldsymbol{\delta} \geq 0$ for all $\boldsymbol{\delta}$, which implies $\nabla f(\boldsymbol{z}) = 0$, the usual definition of a stationary point for smooth functions.

Lemma 3. *The directional derivative of f at $\boldsymbol{z}$ with direction $\boldsymbol{\delta}$ is*

$$f'(\boldsymbol{z}; \boldsymbol{\delta}) = \sum_{i \in \mathcal{I}(\boldsymbol{z})} 2(\boldsymbol{z} - \boldsymbol{x}_i)^\top \boldsymbol{\delta} + \sum_{j \in \mathcal{J}(\boldsymbol{z}; \boldsymbol{\delta})} 2(\boldsymbol{z} - \boldsymbol{x}_j)^\top \boldsymbol{\delta}, \tag{10}$$

where

$$\mathcal{I}(\boldsymbol{z}) = \{i : \|\boldsymbol{x}_i - \boldsymbol{z}\|^2 < w^2\},$$

$$\mathcal{J}(\boldsymbol{z}; \boldsymbol{\delta}) = \{j : \|\boldsymbol{x}_j - \boldsymbol{z}\|^2 = w^2, (\boldsymbol{z} - \boldsymbol{x}_j)^\top \boldsymbol{\delta} < 0\}.$$

Proof. For a smooth point of $\phi(\boldsymbol{z})$, it is easy to see that $\phi'(\boldsymbol{z}; \boldsymbol{\delta}) = 2\boldsymbol{z}^\top \boldsymbol{\delta}$ if $\|\boldsymbol{z}\|^2 < w^2$, and $\phi'(\boldsymbol{z}; \boldsymbol{\delta}) = 0$ if $\|\boldsymbol{z}\|^2 > w^2$. For a non-smooth point when $\|\boldsymbol{z}\|^2 = w^2$, we find its directional derivative by resorting to the definition (8): $\phi'(\boldsymbol{z}; \boldsymbol{\delta})$ equals to $2\boldsymbol{z}^\top \boldsymbol{\delta}$ or 0 depending on whether $\|\boldsymbol{z} + \alpha\boldsymbol{\delta}\|^2$ is less than w^2 or not, when α goes to zero. Since

$$\|\boldsymbol{z} + \alpha\boldsymbol{\delta}\|^2 = \|\boldsymbol{z}\|^2 + 2\alpha\boldsymbol{z}^\top\boldsymbol{\delta} + \alpha^2\|\boldsymbol{\delta}\|^2 = w^2 + 2\alpha\boldsymbol{z}^\top\boldsymbol{\delta} + o(\alpha^2),$$

we see that $\|\boldsymbol{z}+\alpha\boldsymbol{\delta}\|^2 < w^2$ iff $\boldsymbol{z}^\top\boldsymbol{\delta} < 0$. Therefore,

$$\phi'(\boldsymbol{z};\boldsymbol{\delta}) = \begin{cases} 2\boldsymbol{z}^\top\boldsymbol{\delta}, & \|\boldsymbol{z}\| < w^2, \text{ or } \|\boldsymbol{z}\| = w^2 \text{ and } \boldsymbol{z}^\top\boldsymbol{\delta} < 0, \\ 0, & \|\boldsymbol{z}\| > w^2, \text{ or } \|\boldsymbol{z}\| = w^2 \text{ and } \boldsymbol{z}^\top\boldsymbol{\delta} \geq 0. \end{cases} \tag{11}$$

Now using the sum rule for the directional derivative, we conclude that $f'(\boldsymbol{z};\boldsymbol{\delta})$ is as given in (10). □

From the expression of $f'(\boldsymbol{z};\boldsymbol{\delta})$, we can show the following interesting claims:

Proposition 1. *If $\boldsymbol{z}$ is a non-smooth point for $f(\boldsymbol{z})$, then $\boldsymbol{z}$ cannot be a stationary point.*

Proof. From the expression of $\phi'(\boldsymbol{z};\boldsymbol{\delta})$ in (11), we see that the second term in (10) is always ≤ 0. To prove that a non-smooth point cannot be a stationary point, we consider the following two cases:

1. If $\sum_{i\in\mathcal{I}(\boldsymbol{z})} 2(\boldsymbol{z}-\boldsymbol{x}_i) \neq 0$, then there exists a $\boldsymbol{\delta}$ such that $\sum_{i\in\mathcal{I}(\boldsymbol{z})} 2(\boldsymbol{z}-\boldsymbol{x}_i)^\top\boldsymbol{\delta} < 0$, e.g., $\boldsymbol{\delta} = -\sum_{i\in\mathcal{I}(\boldsymbol{z})}(\boldsymbol{z}-\boldsymbol{x}_i)$, and thus $f'(\boldsymbol{z};\boldsymbol{\delta}) < 0$ since the second term in (10) cannot be positive;
2. if $\sum_{i\in\mathcal{I}(\boldsymbol{z})} 2(\boldsymbol{z}-\boldsymbol{x}_i) = 0$, since $\boldsymbol{z}$ is a non-smooth point, there exists a $\boldsymbol{\delta}$ such that $\mathcal{J} \neq \emptyset$, for example by choosing $\boldsymbol{\delta} = -(\boldsymbol{z}-\boldsymbol{x}_j)$ for some j such that $\|\boldsymbol{z}-\boldsymbol{x}_j\|^2 = w^2$, then the second term in (10) is strictly < 0 while the first term $= 0$, therefore $f'(\boldsymbol{z};\boldsymbol{\delta}) < 0$.

To sum up, there always exists a $\boldsymbol{\delta}$ such that $f'(\boldsymbol{z};\boldsymbol{\delta}) < 0$ when $\boldsymbol{z}$ is a non-smooth point for f, therefore such a point cannot be a stationary point. □

Proposition 2. *A point $\boldsymbol{z}_\star$ is a local minimum for $f(\boldsymbol{z})$ iff:*

1. *There does not exist a $\boldsymbol{x}_j$ such that $\|\boldsymbol{x}_j-\boldsymbol{z}\|^2 = w^2$;*
2. *the set $\mathcal{I}(\boldsymbol{z}_\star) = \{i : \|\boldsymbol{x}_i-\boldsymbol{z}_\star\|^2 < w^2\}$ is not empty, and $\boldsymbol{z}_\star = \frac{1}{|\mathcal{I}(\boldsymbol{z}_\star)|}\sum_{i\in\mathcal{I}(\boldsymbol{z}_\star)} \boldsymbol{x}_i$.*

Proof. A local minimum is first of all a stationary point, which, according to Proposition 1, cannot be a non-smooth point for f, therefore there does not exist a $\boldsymbol{x}_j$ such that $\|\boldsymbol{x}_j-\boldsymbol{z}\|^2 = w^2$.

For a smooth point, its gradient and Hessian is given in (6) and (7). Stationarity implies that $\nabla f(\boldsymbol{z}_\star) = 0$, therefore $\boldsymbol{z}_\star = \frac{1}{|\mathcal{I}(\boldsymbol{z}_\star)|}\sum_{i\in\mathcal{I}(\boldsymbol{z}_\star)} \boldsymbol{x}_i$. If $\mathcal{I}(\boldsymbol{z}_\star)$ is not empty, then $\nabla^2 f(\boldsymbol{z}_\star) \succ 0$, and $\boldsymbol{z}_\star$ is a local minimum; otherwise, $f(\boldsymbol{z}_\star) = Mw^2 = \max f(\boldsymbol{z})$, which is a global maximum. □

Algorithm 2 Epanechnikov Mean Shift iterates – Redux

```
Require: $\{\boldsymbol{x}_m\}_{m=1}^M$, $n \in [M]$
initialize $\boldsymbol{z}^{(0)} \leftarrow \boldsymbol{x}_n$
loop
    $\mathcal{I}(\boldsymbol{z}^{(t-1)}) \leftarrow \{i : \|\boldsymbol{x}_i - \boldsymbol{z}^{(t-1)}\|^2 < r\}$
    $\boldsymbol{z}^{(t)} \leftarrow \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|}\sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})} \boldsymbol{x}_i$
    if $\boldsymbol{z}^{(t)} = \boldsymbol{z}^{(t-1)}$ then
        $\mathcal{J}(\boldsymbol{z}^{(t)}) \leftarrow \{j : \|\boldsymbol{x}_i - \boldsymbol{z}^{(t)}\|^2 = r\}$
        if $\mathcal{J}(\boldsymbol{z}^{(t)}) = \emptyset$ then
            return $\boldsymbol{z}^{(t)}$
        else
            sample $j$ from $\mathcal{J}(\boldsymbol{z}^{(t)})$
            $\boldsymbol{z}^{(t)} \leftarrow \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|+1}\left(\boldsymbol{x}_j + \sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})} \boldsymbol{x}_i\right)$
        end if
    end if
    $t \leftarrow t+1$
end loop
```

Convergence of Epanechnikov Mean Shift

Now we study the convergence of the Epanechnikov Mean Shift iterates.

Lemma 4. *Epanechnikov Mean Shift iterates successively minimizes a local upper bound of function $f(\boldsymbol{z})$, therefore the value of $f(\boldsymbol{z})$ is monotonically non-increasing.*

Proof. At a particular point $\widetilde{\boldsymbol{z}}$, define $\overline{f}(\boldsymbol{z}|\widetilde{\boldsymbol{z}})$ as follows:

$$\overline{f}(\boldsymbol{z}|\widetilde{\boldsymbol{z}}) = \sum_{i\in\mathcal{I}(\widetilde{\boldsymbol{z}})} \|\boldsymbol{z}-\boldsymbol{x}_i\|^2 + (M - |\mathcal{I}(\widetilde{\boldsymbol{z}})|)w^2.$$

It is easy to see that

1. $\overline{f}(\boldsymbol{z}|\widetilde{\boldsymbol{z}}) \geq f(\boldsymbol{z})$ for all $\boldsymbol{z}$ and $\widetilde{\boldsymbol{z}}$,
2. $\overline{f}(\widetilde{\boldsymbol{z}}|\widetilde{\boldsymbol{z}}) = f(\widetilde{\boldsymbol{z}})$, and
3. $\frac{1}{|\mathcal{I}(\widetilde{\boldsymbol{z}})|}\sum_{i\in\mathcal{I}(\widetilde{\boldsymbol{z}})} \boldsymbol{x}_i = \arg\min_{\boldsymbol{z}} \overline{f}(\boldsymbol{z}|\widetilde{\boldsymbol{z}})$.

This means that, at iteration t, Algorithm 2 updates $\boldsymbol{z}$ as $\boldsymbol{z}^{(t)} = \arg\min_{\boldsymbol{z}} \overline{f}(\boldsymbol{z}|\boldsymbol{z}^{(t-1)})$. Therefore,

$$f(\boldsymbol{z}^{(t-1)}) = \overline{f}(\boldsymbol{z}^{(t-1)}|\boldsymbol{z}^{(t-1)}) \geq \overline{f}(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)}) \geq f(\boldsymbol{z}^{(t)}),$$

which means the values of $f(\boldsymbol{z})$ obtained by Algorithm 2 form a monotonically non-increasing sequence. □

Lemma 4 gives the Epanechnikov Mean Shift iterates in Algorithm 1 a majorization-minimization interpretation, which guarantees convergence to a stationary point in a lot of cases. Unfortunately none of the existing convergence results applies here—they either require both $f(\boldsymbol{z})$ and $\overline{f}(\boldsymbol{z}|\widetilde{\boldsymbol{z}})$ to be smooth, or more generally $f'(\widetilde{\boldsymbol{z}};\boldsymbol{\delta}) = \overline{f}'(\widetilde{\boldsymbol{z}}|\widetilde{\boldsymbol{z}};\boldsymbol{\delta})$ for all $\widetilde{\boldsymbol{z}}$ and $\boldsymbol{\delta}$ (Razaviyayn, Hong, and Luo 2013). Indeed, it is possible that $\boldsymbol{z}^{(t-1)} = \boldsymbol{z}^{(t)} = \arg\min_{\boldsymbol{z}} \overline{f}(\boldsymbol{z}|\boldsymbol{z}^{(t-1)})$, in which case the algorithm converges, but there exists a $\boldsymbol{x}_j$ such that $\|\boldsymbol{x}_j - \boldsymbol{z}^{(t)}\|^2 = w^2$, meaning it is a non-smooth point, thus cannot be a stationary point according to Proposition 1. Fortunately, we find that this issue can be fixed with negligible extra computations: If such case happens, we only need to sample one $\boldsymbol{x}_j$ such that $\|\boldsymbol{x}_j - \boldsymbol{z}^{(t)}\|^2 = w^2$, and then re-update $\boldsymbol{z}^{(t)}$ as the average of $\boldsymbol{x}_j$ together with all the points in $\mathcal{I}(\boldsymbol{z}^{(t-1)})$. This can be instead interpreted as minimizing the following slightly different upper bound:

$$\begin{aligned}\boldsymbol{z}^{(t)} &= \arg\min_{\boldsymbol{z}} \overline{f}_\sharp(\boldsymbol{z}|\boldsymbol{z}^{(t-1)}) \\ &= \|\boldsymbol{z}-\boldsymbol{x}_j\|^2 + \sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\|\boldsymbol{z}-\boldsymbol{x}_i\|^2 + (M-|\mathcal{I}(\boldsymbol{z}^{(t-1)})|-1)w^2.\end{aligned}$$

This elaborated procedure is fleshed out in Algorithm 2. We show that Algorithm 2 provably finds a local optimum in a finite number of iterations.

Lemma 5. *The value of $f(\boldsymbol{z})$ obtained by Algorithm 2 is strictly decreasing, unless $\boldsymbol{z}^{(t)} = \boldsymbol{z}^{(t-1)}$.*

Proof. If $\boldsymbol{z}^{(t-1)} \neq \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|}\sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\boldsymbol{x}_i$, then we update $\boldsymbol{z}^{(t)} = \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|}\sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\boldsymbol{x}_i$, and

$$\begin{aligned}&f(\boldsymbol{z}^{(t-1)}) - f(\boldsymbol{z}^{(t)}) \\ &\geq \overline{f}(\boldsymbol{z}^{(t-1)}|\boldsymbol{z}^{(t-1)}) - \overline{f}(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)}) \\ &= \sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\|\boldsymbol{z}^{(t-1)} - \boldsymbol{x}_i\|^2 - \sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\|\boldsymbol{z}^{(t)} - \boldsymbol{x}_i\|^2 \\ &= \sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\Big(\|\boldsymbol{z}^{(t-1)}\|^2 - 2\boldsymbol{x}_i^\top\boldsymbol{z}^{(t-1)} - \|\boldsymbol{z}^{(t)}\|^2 + 2\boldsymbol{x}_i^\top\boldsymbol{z}^{(t)}\Big) \\ &= |\mathcal{I}(\boldsymbol{z}^{(t-1)})|\Big(\|\boldsymbol{z}^{(t-1)}\|^2 - 2\boldsymbol{z}^{(t)\top}\boldsymbol{z}^{(t-1)} - \|\boldsymbol{z}^{(t)}\|^2 + 2\|\boldsymbol{z}^{(t)}\|^2\Big) \\ &= \Big|\mathcal{I}(\boldsymbol{z}^{(t-1)})\Big|\Big\|\boldsymbol{z}^{(t-1)} - \boldsymbol{z}^{(t)}\Big\|^2 > 0.\end{aligned}$$

Similarly, if $\boldsymbol{z}^{(t-1)} = \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|}\sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\boldsymbol{x}_i$, we have

$$\begin{aligned}&f(\boldsymbol{z}^{(t-1)}) - f(\boldsymbol{z}^{(t)}) \\ &\geq \overline{f}_\sharp(\boldsymbol{z}^{(t-1)}|\boldsymbol{z}^{(t-1)}) - \overline{f}_\sharp(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)}) \\ &= \Big(|\mathcal{I}(\boldsymbol{z}^{(t-1)})| + 1\Big)\Big\|\boldsymbol{z}^{(t-1)} - \boldsymbol{z}^{(t)}\Big\|^2 \\ &= \Big(|\mathcal{I}(\boldsymbol{z}^{(t-1)})| + 1\Big)\Bigg\|\frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|}\sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\boldsymbol{x}_i \\ &\qquad - \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|+1}\Bigg(\boldsymbol{x}_j + \sum_{i\in\mathcal{I}(\boldsymbol{z}^{(t-1)})}\boldsymbol{x}_i\Bigg)\Bigg\|^2 \\ &= \frac{1}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|+1}\Big\|\boldsymbol{z}^{(t-1)} - \boldsymbol{x}_j\Big\|^2 \\ &= \frac{w^2}{|\mathcal{I}(\boldsymbol{z}^{(t-1)})|+1} > 0.\end{aligned}$$

□

Theorem 1. *Algorithm 2* ***terminates*** *at a local optimum of* (5) *in a finite number of iterations.*

Proof. We first prove that Algorithm 2 terminates at a local optimum of (5). The loss function $f(\boldsymbol{z})$ is bounded from below, and using Algorithm 2, it is monotonically non-increasing (cf. Lemma 4), so it converges to a certain value. Lemma 5 further shows that $f(\boldsymbol{z})$ strictly decreases unless $\boldsymbol{z}^{(t)} = \boldsymbol{z}^{(t-1)}$, in which case $\boldsymbol{z}^{(t)}$ cannot be a non-smooth point as we showed in the proof of Lemma 5. Notice that throughout the iterations $\mathcal{I}(\boldsymbol{z}^{(t)})$ cannot be empty, because otherwise $f(\boldsymbol{z})$ takes the maximum value, but since we start with $\boldsymbol{z}^{(0)} = \boldsymbol{x}_m$, $f(\boldsymbol{z}^{(0)}) < \max f(\boldsymbol{z})$. Invoking Proposition 2, we conclude that Algorithm 2 terminates at a local optimum.

Now suppose Algorithm 2 terminates in T number of iterations, we show that T can be upperbounded by a finite number that only depends on the data set $\{\boldsymbol{x}_m\}$ and the bandwidth w that we choose. From the proof of Lemma 5, we get the (very loose) inequality

$$f(\boldsymbol{z}^{(t-1)}) - f(\boldsymbol{z}^{(t)}) \geq \|\boldsymbol{z}^{(t-1)} - \boldsymbol{z}^{(t)}\|^2. \tag{12}$$

Summing up both sides for $t = 1, ..., T$, we have

$$f(\boldsymbol{z}^{(0)}) - f(\boldsymbol{z}^{(T)}) \geq \sum_{t=1}^{T}\|\boldsymbol{z}^{(t-1)} - \boldsymbol{z}^{(t)}\|^2. \tag{13}$$

We have shown that each of the terms on the right-hand-side is positive, unless the algorithm has terminated. If we can further find a quantity $\lambda > 0$ such that

$$\|\boldsymbol{z}^{(t-1)} - \boldsymbol{z}^{(t)}\|^2 \geq \lambda > 0, \forall\, t = 1, ..., T, \tag{14}$$

then we can easily conclude that

$$T \leq \frac{f(\boldsymbol{z}^{(0)}) - f(\boldsymbol{z}^{(T)})}{\lambda}, \tag{15}$$

which is a finite number.

To find such a λ, we note that each $\boldsymbol{z}^{(t)}$ is the average of a non-empty subset of points from the data set $\{\boldsymbol{x}_m\}$, which are all the points that can be enclosed in a Euclidean ball with radius w (except for $\boldsymbol{z}^{(0)}$, which is simply one data point). Define

$$\mathcal{S} = \{\mathcal{I}(\boldsymbol{z}) \cup \mathcal{J}(\boldsymbol{z};\boldsymbol{\delta}) : \forall\, \boldsymbol{z}, \boldsymbol{\delta} \in \mathbb{R}^d\},$$

where $\mathcal{I}(\boldsymbol{z})$ and $\mathcal{J}(\boldsymbol{z};\boldsymbol{\delta})$ are as defined in Lemma 3. We know that $|\mathcal{S}| < 2^M$, since it is a union of subsets of $\{\boldsymbol{x}_m\}$, and there can be at most $2^M - 1$ non-empty subsets of $\{\boldsymbol{x}_m\}$. Then we define

$$\lambda = \min_{\substack{\mathcal{K},\mathcal{L}\in\mathcal{S}\\ \mathcal{K}\neq\mathcal{L}}}\left\|\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\boldsymbol{x}_k - \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\boldsymbol{x}_\ell\right\|^2, \tag{16}$$

which exists since $\mathcal{S}$ is a finite set, and it satisfies (14). This λ is also strictly positive: If

$$\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\boldsymbol{x}_k = \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\boldsymbol{x}_\ell, \tag{17}$$

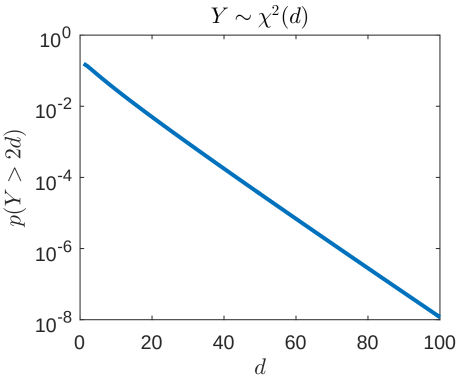

Figure 2: Probability that $Y > 2d$, for $Y \sim \chi^2(d)$.

the distance between this point and every point in either $\mathcal{K}$ or $\mathcal{L}$ is no greater than w. According to the construction of $\mathcal{S}$, (17) implies $\mathcal{K} = \mathcal{L}$, contradicting $\mathcal{K} \neq \mathcal{L}$ in (16).

To sum up, we have shown that there exists $\lambda > 0$ that satisfies (14), thus (15) holds, meaning Algorithm 2 terminates in a finite number of iterations. □

We remark that (15) is a gross over-estimate of the number of iterations. We omitted a rather big scaling factor on the right-hand-side of (12), for the sake of simplicity in (13). The point is to show that T can indeed be bounded by a finite number. In practice, our observation is that Epanechnikov Mean Shift terminates in a very smaller number of iterations, usually less than 10.

As we can see, even though we are trying to optimize a non-convex and non-smooth function, we obtain a very strong convergence result that Epanechnikov Mean Shift reaches (not *approaches*) a local optimum in a finite number of iterations. From the proof one can see that the non-smoothness of the Epanechnikov kernel actually helps obtaining such a nice convergence property. It has a very different flavor as the smooth optimization based analyses. For example, the work in (Carreira-Perpiñán 2007) uses a smooth Gaussian kernel, and the analysis ends up with an asymptotic convergence and a sub-linear rate in the worst case.

Mean Shift Deflation for Non-overlapping Spherical Gaussian Mixtures

The main computation bottleneck for Mean Shift is obviously the fact that the Mean Shift iterates (Algorithm 2) is run at *every* data point, which is potentially a large set. In this section we provide a heuristic to avoid this computation bottleneck, under the probabilistic generative model that each cluster is generated from a spherical Gaussian with variance $\sigma^2\boldsymbol{I}$, and they are non-overlapping.

Suppose the set of data points $\{\boldsymbol{x}_m\}_{m=1}^M \subseteq \mathbb{R}^d$ comes from a mixture of Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I})$, $\forall k = 1, ..., K$, with different means $\{\boldsymbol{\mu}_k\}_{k=1}^K \subseteq \mathbb{R}^d$ but the same covariance matrix $\sigma^2\boldsymbol{I}$. For a specific data point $\boldsymbol{x}_m$ coming from $\mathcal{N}(\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I})$, the random variable $Y = \|\boldsymbol{x}_m - \boldsymbol{\mu}_k\|^2/\sigma^2$ follows the χ^2-distribution with d degrees of freedom, denoted as $\chi^2(d)$.

An interesting property of the χ^2-distribution is that the probability that $Y > \gamma d$ becomes almost negligible for large d and $\gamma > 1$. As an example, Figure 2 shows the probability that $Y > 2d$, as d increases. This can be seen from the Chernoff bound on the tail probability of the χ^2-distribution (Dasgupta and Gupta 2003): for $\gamma > 1$, we have that $\Pr(Y > \gamma d) \leq (\gamma e^{1-\gamma})^{d/2}$. It can be easily shown that $\gamma e^{1-\gamma} < 1$ when $\gamma > 1$, implying that $\Pr(Y > \gamma d)$ goes to zero *at least exponentially* as d increases.

Algorithm 3 Epanechnikov Mean Shift deflation

```
Require: $\{\boldsymbol{x}_m\}_{m=1}^M$, $w^2 = 2d\sigma^2$
$\mathcal{M} \leftarrow \{1, 2, ..., M\}$, $k \leftarrow 1$
while $\mathcal{M} \neq \emptyset$ do
    sample $m$ from $\mathcal{M}$
    run Algorithm 2 initialized at $\boldsymbol{x}_m$, outputs $\boldsymbol{\mu}_k$
    declare $\{\boldsymbol{x}_i\}_{i \in \mathcal{I}(\boldsymbol{\mu}_k)}$ as cluster $k$
    $\mathcal{M} \leftarrow \mathcal{M} \setminus \mathcal{I}(\boldsymbol{\mu})$
    $k \leftarrow k + 1$
end while
```

This observation inspires us to use $\sqrt{2d}\sigma$ as the bandwidth, since a Euclidean ball with radius $\sqrt{2d}\sigma$ encloses almost all points coming from $\mathcal{N}(\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I})$ if it is centered at $\boldsymbol{\mu}_k$. Furthermore, if $\min \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell\| > 2\sqrt{d}\sigma$, the Gaussian components are non-overlapping, thus the ball centered at $\boldsymbol{\mu}_k$ will *only* contain points coming from $\mathcal{N}(\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I})$. Therefore, once we find one local optimum of $\widehat{p}(\boldsymbol{z})$ that is presumably close to a $\boldsymbol{\mu}_k$, we can safely group all data points that are within radius $\sqrt{2d}\sigma$ from $\boldsymbol{\mu}_k$ and declare them as a cluster. This idea leads to the Epanechnikov Mean Shift *deflation* shown in Algorithm 3.

As shown in Figure 1, this simple procedure obtains extremely good clustering performance, while reducing the computational complexity down to even smaller than that of Lloyd's K-means algorithm. Notice that the bandwidth w is still estimated via leave-one-out cross-validation. Under strong generative models as in this case, however, we do observe that the estimated bandwidth w is very close to $\sqrt{2d}\sigma$.

Conclusion

We study the Epanechnikov Mean Shift algorithm, which is observed to work well but lacked theoretical analysis on its performance as of this writing. After in-depth study on estimated density $\widehat{p}(\boldsymbol{z})$, with particular focus on its non-smoothness, we fixed an issue that could potentially affect its convergence, and showed that the Epanechnikov Mean Shift iterate terminates at a local optimum within finite number of iterations. A deflation-based variant of Epanechnikov Mean Shift is also proposed to avoid initializing the iterates at every data point, which reduces the computation considerably, and maintains good clustering performance under non-overlapping spherical Gaussian mixture assumptions.

Acknowledgments

This work was supported in part by the National Science Foundation (NSF) under grants IIS-1247632, IIS-1447788, and ECCS-1608961.

References

Aloise, D.; Deshpande, A.; Hansen, P.; and Popat, P. 2009. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning* 75(2):245–248.

Arora, S., and Kannan, R. 2001. Learning mixtures of arbitrary Gaussians. In *ACM Symposium on Theory of Computing (STOC)*, 634–644.

Arthur, D., and Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035. Society for Industrial and Applied Mathematics.

Belkin, M., and Sinha, K. 2010. Polynomial learning of distribution families. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 103–112.

Bertsekas, D. P. 1999. *Nonlinear programming*. Athena Scientific.

Brubaker, S. C., and Vempala, S. S. 2008. Isotropic PCA and affine-invariant clustering. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 551–560.

Carreira-Perpiñán, M. Á. 2007. Gaussian mean-shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(5):767–776.

Cheng, Y. 1995. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(8):790–799.

Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5):603–619.

Dasgupta, S., and Freund, Y. 2009. Random projection trees for vector quantization. *IEEE Transactions on Information Theory* 55(7):3229–3242.

Dasgupta, S., and Gupta, A. 2003. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms* 22(1):60–65.

Dasgupta, S., and Schulman, L. J. 2000. A Two-Round Variant of EM for Gaussian Mixtures. In *Uncertainty in Artificial Intelligence*, 152–159.

Dasgupta, S. 1999. Learning mixtures of Gaussians. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 634–644.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 1–38.

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.

Epanechnikov, V. A. 1969. Non-parametric Estimation of A Multivariate Probability Density. *Theory of Probability and Its Applications* 14(1):153—-158.

Fukunaga, K., and Hostetler, L. D. 1975. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory* 21(1):32–40.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: A review. *ACM Computing Surveys (CSUR)* 31(3):264–323.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1-2):83–97.

Lloyd, S. P. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory* 28(2):129–137.

Moitra, A., and Valianty, G. 2010. Settling the polynomial learnability of mixtures of Gaussians. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 93–102.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 849–856.

Razaviyayn, M.; Hong, M.; and Luo, Z.-Q. 2013. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization* 23(2):1126–1153.

Scott, D. W. 2015. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.

Vempala, S., and Wang, G. 2004. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences* 68(4):841–860.