Mean Shift: An Information Theoretic Perspective

Sudhir Rao, *Student Member, IEEE,* and José C. Príncipe, *Fellow, IEEE*

***Abstract*— This paper develops a new understanding of mean shift algorithms from an information theoretic perspective. We show that the Gaussian Blurring Mean Shift (GBMS) directly minimizes the Renyi's quadratic entropy of the dataset and hence is unstable by definition. Further, its stable counterpart, the Gaussian Mean Shift (GMS), minimizes the Renyi's "cross" entropy where the local stationary solutions are modes of the dataset. By doing so, we aptly answer the question "What does mean shift algorithms optimize?", thus highlighting naturally the properties of these algorithms. A consequence of this new understanding is the superior performance of GMS over GBMS which we show in a wide variety of applications ranging from mode finding to clustering and image segmentation.**

***Index Terms*— Mean shift, information theoretic learning, Renyi's entropy.**

I. Introduction

LET us consider a dataset $X = (x_i)_{i=1}^N \in \mathbf{R}^d$ with independent and identically distributed (iid) samples. Using the non-parametric method of Parzen windowing, the probability density estimate is given by

$$p_{X,\sigma}(x) = \frac{1}{N}\sum_{i=1}^{N} G_\sigma(x - x_i), \tag{1}$$

where $G_\sigma(t) = e^{-\frac{t^2}{2\sigma^2}}$ is a Gaussian kernel with bandwidth $\sigma > 0$. In order to find the modes of the pdf we rearrange the stationary point equation $\nabla p_{X,\sigma}(x) = 0$ into an iterative fixed point scheme

$$x^{(\tau+1)} = m(x^{(\tau)}) = \frac{\sum_{i=1}^{N} G_\sigma(x - x_i)x_i}{\sum_{i=1}^{N} G_\sigma(x - x_i)} \tag{2}$$

Note that the expression $m(x)$ is the sample mean of all the samples x_i weighted by the kernel centered at x. Thus the term $m(x) - x$ was coined "mean shift" by Fukunaga and Hostetler in their landmark paper [1]. Given an initial dataset $X^{(0)} = X_o$ and using (2), we successively "blur" the dataset X_o to produce datasets $X^{(1)}, X^{(2)} \ldots X^{(\tau)}$. As the new datasets are produced we forget the previous one which gives rise to the blurring process. It was Cheng [2] who first pointed out this and renamed the fixed point update (2) as blurring mean shift.

This successive blurring made the data to collapse rapidly and hence made the algorithm unstable. In his 1995 paper, which sparked renewed interest in mean shift, Cheng proposed a modification in which two different datasets would be maintained namely X and X_o. The dataset X would be initialized to X_o as $X^{(0)} = X_o$. At every iteration, a new dataset $X^{(\tau+1)}$ is produced by comparing the present dataset $X^{(\tau)}$ with X_o. Throughout this process X_o is fixed and kept constant. This stable fixed point update was called mean shift algorithm and is summarized in (3) where $(x_{oi})_{i=1}^{N_o}$ are samples of the original dataset X_o.

$$x^{(\tau+1)} = m(x^{(\tau)}) = \frac{\sum_{i=1}^{N_o} G_\sigma(x - x_{oi})x_{oi}}{\sum_{i=1}^{N_o} G_\sigma(x - x_{oi})}. \tag{3}$$

To be consistent with the existing mean shift literature, we call these algorithms Gaussian blurring mean shift (GBMS) and Gaussian mean shift (GMS) respectively indicating the use of Gaussian kernel specifically.

Recent advancements in Gaussian mean shift has made it increasing popular in image processing and vision communities. In particular, the mean shift vector of GMS has been shown to always point in the direction of normalized density gradient [2]. Since points lying in low density region have small value of $p(x)$, the normalized gradient at these points have large value. This helps the samples to quickly move from low density regions toward the modes. On the other hand, due to relatively high value of $p(x)$ near the mode, the steps are highly refined around this region. This adaptive nature of step size gives GMS a significant advantage over traditional gradient based algorithms where step size selection is well known problem.

A rigorous proof of stability and convergence of GMS was given by Comaniciu *et al.* [3] where he proved that the sequence generated by (3) is a Cauchy sequence that converges due to the monotonic increasing sequence of the pdfs estimated at these points. Further the trajectory is always smooth in the sense that the consecutive angles between mean shift vectors is always between $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Carreira-Perpiñán [4] also showed that GMS is an Expectation-Maximization (EM) algorithm and thus has a linear convergence rate.

Due to these interesting and useful properties, GMS has been successfully applied in low level vision tasks like image segmentation and discontinuity preserving smoothing [3] as well as in high level vision tasks like appearance based clustering [5] and real-time tracking of non rigid objects [6]. Carreira-Perpiñán [7] used mean shift for mode finding in mixture of Gaussian distributions. The connection to Nadarayana-Watson estimator from kernel regression and the robust M-estimators of location has been thoroughly explored by Comaniciu *et al.* [3]. With just a single parameter to control the scale of analysis, this simple non-parametric iterative procedure has become particularly attractive and suitable for wide range of applications.

On the other hand, the understanding of GBMS algorithm remains poor since this concept first appeared in [1]. Apart from the preliminary work done in [2], the only other notable contribution which we are aware of was recently made by Carreira-Perpiñán. In his paper [8], the author showed that GBMS has a cubic convergence rate and to overcome its instability, developed a new stopping criterion. By removing the redundancy among points which have already merged, an accelerated GBMS was developed which was $2\times - 4\times$ faster[1].

Sudhir Rao is with Computational Neuroengineering Laboratory (CNEL), Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (email: {sudhir, principe}@cnel.ufl.edu).

[1]Note that this can also be done for GMS algorithm

In spite of these achievements, little progress has been made to understand mean shift algorithms theoretically. For example, the question still unanswered is "what do these algorithms optimize?". Fashing *et al.* [9] showed mean shift as quadratic bound maximization but the analysis is indirect and the scope limited. Further, the implications and instability of GBMS is least understood. It is also not clear what changes are incurred when going from GBMS to GMS and vice versa. Cheng et al. [2] tried to address this issue with various postulates and optimization concepts making the analysis very complex. In this paper we successfully answer some of these issues. By bringing in fresh perspective to these algorithms from information theoretic point of view we simplify greatly the understanding of these algorithms.

In next section we introduce information theoretic concepts. Section 3 explores the connection between mean shift algorithms and Renyi's entropy and its implications. We show the instability of GBMS in mode finding leading to its poor performance compared to GMS in clustering and image segmentation problems in section 4 and finally we conclude with discussion in section 5.

II. Information Theoretic Learning (ITL)

Let $X = (x_i)_{i=1}^N \in \mathbf{R}^d$ be a random variable with independent and identically distributed samples. The non-parametric density estimator using Parzen windowing technique is given by

$$p_{X,\Sigma}(x) = \frac{1}{N}\sum_{i=1}^{N} K_\Sigma(x - x_i), \tag{4}$$

where K_Σ is a kernel with covariance matrix Σ. Although in principle a full covariance matrix can be used, for simplicity and ease of estimation, we will only consider spherical covariance of the form $\Sigma = \sigma^2 I$ for which a number of well established techniques exists from kernel density estimation literature [10].

Throughout this paper we use the Gaussian kernel. The advantage of this kernel selection is two-folded. First, it is a smooth, continuous and infinitely differentiable kernel and has been shown to outperform other kernels in applications where mean shift has been employed [3]. Second, the Gaussian kernel is the only kernel with a very special property that the integral of the product of two Gaussian functions is exactly equal to another Gaussian function with variance equal to the sum of the variances of the original Gaussian functions. This property forms the key in developing a non-parametric estimator for Renyi's entropy.

Renyi's quadratic entropy is defined as [11]

$$H(X) = -\log\Big(\int p^2(x)dx\Big). \tag{5}$$

Substituting the Parzen estimate of $p(x)$ using a Gaussian kernel and spherical covariance $\Sigma = \sigma_X^2 I$ and using the property of Gaussian kernel stated above we get a non-parametric entropy estimator as shown below.

$$\begin{aligned} H(X) &= -\log(V(X)) \\ V(X) &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_\sigma\big(x_i - x_j\big), \end{aligned} \tag{6}$$

where $\sigma^2 = 2\sigma_X^2$. Notice the argument of the Gaussian kernel which considers all possible pairs of samples. The idea of regarding the samples as information particles was first introduced by Príncipe *et al.* and collaborators [12], [13] upon realizing that these samples interact with each other through laws that resembled the potential fields and their associated forces in physics.

Since the log is a monotonic function, any optimization based on $H(X)$ can be translated into optimization of argument of the log which we denote by $V(X)$ and call the information potential of the samples. We can consider this quantity as a sum of contributions from each particle x_i given by

$$V(x_i) = \frac{1}{N^2}\sum_{j=1}^{N} G_\sigma\big(x_i - x_j\big). \tag{7}$$

Note that $V(x_i)$ is the potential field over the space of the samples, with an interaction law given by the kernel shape. The derivative of this contribution with respect to the value of the sample is given by

$$\frac{\partial}{\partial x_i}V(x_i) = \frac{1}{N^2}\sum_{j=1}^{N} G_\sigma\big(x_i - x_j\big)\big(\frac{x_j - x_i}{\sigma^2}\big). \tag{8}$$

We can regard this derivative as a contribution of derivatives due to all other samples and denoting the contribution by sample x_j with $F(x_i \mid x_j)$ and overall derivative with respect to x_i with $F(x_i)$, we get

$$\begin{aligned} F(x_i) &= \frac{\partial}{\partial x_i}V(x_i) = \sum_{j=1}^{N} F(x_i \mid x_j) \\ F(x_i \mid x_j) &= \frac{1}{N^2} G_\sigma\big(x_i - x_j\big)\big(\frac{x_j - x_i}{\sigma^2}\big). \end{aligned} \tag{9}$$

$F(x_i \mid x_j)$ is the information force exerted by particle x_j on particle x_i whereas $F(x_i)$ is the net force acting on sample x_i.

This idea of interaction between samples of the same dataset can be extended to quantify interactions between two different datasets. Let $X = (x_i)_{i=1}^N$ and $Y = (y_j)_{j=1}^M$ be iid samples from two different random variables in $\mathbf{R}^d$. Let $p_{X,\sigma_X}(x)$ and $p_{Y,\sigma_Y}(y)$ denote the pdfs of X and Y estimated non-parametrically with Gaussian kernel and covariance matrix $\sigma_X^2 I$ and $\sigma_Y^2 I$ respectively. Then, we define the Renyi's "cross" entropy between two pdfs as

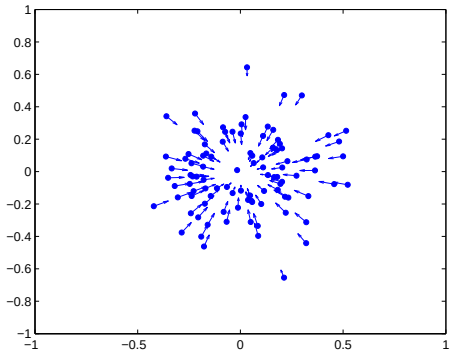
$$H(X;Y) = -\log\Big(\int p_X(t)p_Y(t)dt\Big). \tag{10}$$

Substituting the Parzen estimates of pdfs of X and Y yields Renyi's cross information potential given by

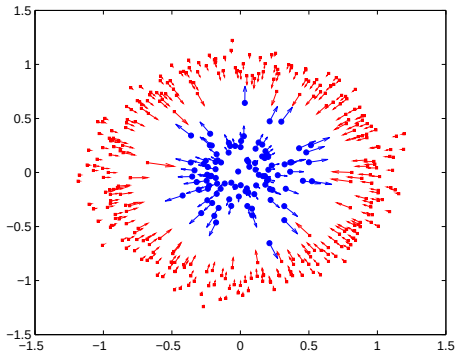
$$\begin{aligned} V(X;Y) = E_{p_Y}[p_X(X)] &= \int p_X(t)p_Y(t)dt \\ &= \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M} G_\sigma\big(x_i - y_j\big), \end{aligned} \tag{11}$$

where $\sigma^2 = \sigma_X^2 + \sigma_Y^2$. The information potential and force experienced by particle $x_i \in X$ due to all particles of dataset Y is shown in (12) where $F(x_i \mid y_j)$ is the "cross" information force exerted by particle y_j on particle x_i. Similarly, one can easily derive the potential and force experienced by $y_i \in Y$ due to all particles of dataset X by simply interchanging $M \leftrightarrow N$ and $x \leftrightarrow y$ in (12). Fig. 1 summarizes these concepts neatly.

These ideas lie at the heart of information theoretic learning (ITL) [12]. By playing directly with pdf of the data and estimating the entropy non-parametrically, ITL effectively goes beyond the second order statistics. The result is new cost functions that directly manipulate information, thus bringing in powerful techniques and applications in adaptive systems [13] and machine learning [14], [15].



(a) Information force within a dataset



(b) "Cross" information force between two datasets

Fig. 1. Concept of information force arising due to H(X) and H(X;Y)

$$V(x_i;Y) = \frac{1}{MN}\sum_{j=1}^{M} G_\sigma(x_i - y_j)$$
$$F(x_i;Y) = \frac{\partial}{\partial x_i}V(x_i;Y) = \sum_{j=1}^{M} F(x_i \mid y_j) \qquad (12)$$
$$= \frac{1}{MN}\sum_{j=1}^{M} G_\sigma(x_i - y_j)\left(\frac{y_j - x_i}{\sigma^2}\right)$$

III. Mean shift and Renyi's entropy

We now develop the connection between mean shift algorithms and Renyi's entropy. Consider an original dataset $X_o = (x_o)_{i=1}^{N_o} \in \mathbf{R}^d$ with iid samples. This dataset is kept fixed throughout the experiment. Let us define another dataset $X = (x)_{i=1}^{N} \in \mathbf{R}^d$ with initialization $X = X_o$ and $\sigma_X = \sigma_{X_o}$. With this setup, consider the following cost function.

$$J(X) = \min_X \; H(X) = \min_X \; -log\big(V(X)\big) \qquad (13)$$

Notice that X is the variable which evolves over time and hence appears as argument of the cost function. Since log is a monotonous function we can redefine $J(X)$ as

$$J(X) = \max_X \; V(X) = \max_X \; \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_\sigma(x_i - x_j).$$

Differentiating $J(X)$ with respect to $x_{k=\{1,2,\ldots,N\}} \in X$ and equating it to zero gives

$$2\; F(x_k) = \frac{2}{N^2}\sum_{j=1}^{N} G_\sigma(x_k - x_j)\left(\frac{x_j - x_k}{\sigma^2}\right) = 0$$

$F(x_k)$ is the information force acting on particle x_k due to all other samples within the dataset X. Thus we would like to evolve this dataset such that the samples reach an equilibrium position with net force acting on each sample equal to zero. Rearranging the above equation gives us the fixed point update rule for each x_k as shown below.

$$x_k^{(\tau+1)} = m(x_k^{(\tau)}) = \frac{\sum_{j=1}^{N} G_\sigma(x_k - x_j)x_j}{\sum_{j=1}^{N} G_\sigma(x_k - x_j)} \qquad (14)$$

Comparing this to (2) we see that this is exactly equal to GBMS algorithm. Thus GBMS minimizes the overall Renyi's quadratic entropy of the dataset. Since the only stationary solution of this is a single point, we conclude immediately that GBMS is unstable. With X initialized to the original dataset X_o, successive iterations of this fixed point algorithm would "blur" the dataset ultimately giving us a single point which is useless.

GBMS has been used to find the modes of the data and further extended to clustering and image segmentation applications [2], [8]. We argue (supported by our experiments) that this is true only when the modes are far apart compared to the kernel size. Further, modes are neither stationary nor saddle points of cost function $H(X)$ which GBMS minimizes. Thus any stopping criteria for this algorithm would at most be heuristic and there is no guarantee that all the modes will ever be found.

We can rectify this deficiency by making a slight modification to the cost function. Instead of minimizing Renyi's quadratic entropy we minimize Renyi's cross entropy $H(X;X_o)$ (or maximize $V(X;X_o)$).

$$J(X) = \max_X \; V(X;X_o)$$
$$= \max_X \; \frac{1}{NN_o}\sum_{i=1}^{N}\sum_{j=1}^{N_o} G_\sigma(x_i - x_{oj}) \qquad (15)$$

Differentiating $J(X)$ with respect to $x_{k=\{1,2,\ldots,N\}} \in X$ and equating it to zero gives

$$\frac{\partial}{\partial x_k}J(X) = 2\; F(x;X_o) = 0$$

Thus in this scenario, the particles of dataset X move under the influence of the "cross" information force exerted by samples from dataset X_o. The fixed point update would then be

$$x_k^{(\tau+1)} = m(x_k^{(\tau)}) = \frac{\sum_{j=1}^{N} G_\sigma(x_k - x_{oj})x_{oj}}{\sum_{j=1}^{N} G_\sigma(x_k - x_{oj})} \qquad (16)$$

Indeed, this is the GMS update equation as shown in (3). By minimizing $H(X;X_o)$, GMS evolves the dataset X and at the same time keeps in "memory" the original dataset X_o. Since $F(x;X_o) \propto \nabla p_{X_o,\sigma}(x)$, the result is movement of the samples $x_{k=\{1,2,\ldots,N\}} \in X$ toward the modes of the dataset X_o (with kernel size σ)[2] where $F(x;X_o) = 0$.

[2]Note that in practice, we never have to select σ_X or σ_{X_o}. Given a dataset X_o, we estimate the kernel size σ and directly compute the entropy. In case of GMS, mean shift with this kernel size would then track the modes of the pdf $p_{X_o,\sigma}(x)$.

Theorem 1: With X initialized to X_o in GMS, $H(X;X_o)$ reaches its local minimum at the fixed points of (16).

Proof: Using (16), the mean shift vector in GMS at iteration τ would be

$$\begin{aligned} x^{\tau+1} - x^{\tau} &= m(x^{\tau}) - x^{\tau} \\ &= \frac{\frac{1}{N_o}\sum_{j=1}^{N_o} G_\sigma(x - x_{oj})(x_{oj} - x)}{\frac{1}{N_o}\sum_{j=1}^{N_o} G_\sigma(x - x_{oj})} \\ &= \frac{1}{2}\sigma^2 \frac{\nabla_x p_{X_o,\sigma}(x)}{p_{X_o,\sigma}(x)} \\ &= \frac{1}{2}\sigma^2 \nabla_x \log(p_{X_o,\sigma}(x)). \end{aligned}$$

Thus the samples move in the direction of normalized density gradient with increasing density values. Each sample converges to that mode to whose convex hull it belongs [3]. Let $s_{l=\{1,2,\ldots,L\}}$ be the modes of $p_{X_o,\sigma}(x)$. Associate each $x_{i=\{1,2,\ldots,N\}} \in X$ with its corresponding mode $s_{i^*}, i^* \in \{1,2,\ldots,L\}$ to which it converges. Then,

$$\begin{aligned} V(X;Xo) &= \frac{1}{NN_o}\sum_{i=1}^{N}\sum_{j=1}^{N_o} G_\sigma(x_i - x_{oj}) \\ &= \frac{1}{N}\sum_{i=1}^{N} p_{X_o,\sigma}(x_i) \\ &\leq \frac{1}{N}\sum_{i=1}^{N} p_{X_o,\sigma}(s_{i^*}) \\ &\leq \max_{s_l}\ p_{X_o,\sigma}(s_l). \end{aligned}$$

Since $V(X;Xo) = \frac{1}{N}\sum_{i=1}^{N} p_{X_o,\sigma}(s_{i*})$ at the fixed points (modes) of (16) and $H(X;X_o) = -log\big(V(X;Xo)\big)$, $H(X;X_o)$ reaches its local minimum starting with initialization $H(X;X_o) = H(X_o)$. ∎

A. Stopping Criterion

1) GMS: Stopping the GMS algorithm to find the modes is very simple. Since samples move in the direction of normalized gradient toward the modes which are fixed points of (16), the average distance moved by samples becomes smaller over subsequent iterations. By setting a *tol* level on this quantity to a low value we can get the modes as well as stop GMS from running unnecessarily. This is summarized in (17).

$$\begin{aligned} \text{Stop when}\ \ \frac{1}{N}\sum_{i=1}^{N} d^{(\tau)}(x_i) < tol\ \ \text{where} \\ d^{(\tau)}(x_i) = \|x_i^{(\tau)} - x_i^{(\tau-1)}\| \end{aligned} \tag{17}$$

2) GBMS: As stated earlier, modes are not the solution of GBMS fixed point update equation and hence GBMS cannot be used to find them. But assume that the modes are far apart compared to kernel size. In such cases, there generally seems to be two distinct phases of convergence. In the first phase, the points quickly collapse to their respective modes while the modes move very slowly towards each other. In the second phase, the modes start merging and ultimately yield a single point. If the algorithm can be stopped after the first phase then it could be used in applications like clustering where the exact position of modes is not important, although any such stopping criterion would at most be heuristic. Of course the stopping criterion (17) cannot be used unless we hand-pick the *tol* level since the average distance moved by the particles never settles down until all of them have merged.

The above assumption was effectively used to formulate a stopping criterion by Carreira-Perpiñán [8]. In phase 2, $d^{(\tau)} = \{d^{(\tau)}(x_i)\}_{i=1}^{N}$ takes on at most K different values (for K modes). Binning $d^{(\tau)}$ using large number of bins gives us the histogram which has K or fewer non empty bins. Since entropy does not depend on exact location of the bins, its value does not change and can be used to stop the algorithm as shown in (18).

$$\left| H_s(d^{(\tau+1)}) - H_s(d^{(\tau)}) \right| < 10^{-8} \tag{18}$$

where $H_s(d) = -\sum_{i=1}^{B} f_i \log f_i$ is the Shannon entropy, f_i is the relative frequency of bin i and the bins span the interval $[0, max(d)]$. The number of bins B was selected as $B = 0.9N$.

It is clear that there is no guarantee that we would find all the modes using this rule. Further, the assumption used in developing this criterion does not hold true in many practical scenarios as will be shown in our experiments.

IV. Applications

We corroborate this new understanding through a detailed set of experiments. We first start with the mode finding ability of GBMS and compare it with its stable counterpart, the GMS algorithm. We then extend this to clustering and ultimately apply it to segment real images where the implications of the instability of GBMS become clear.

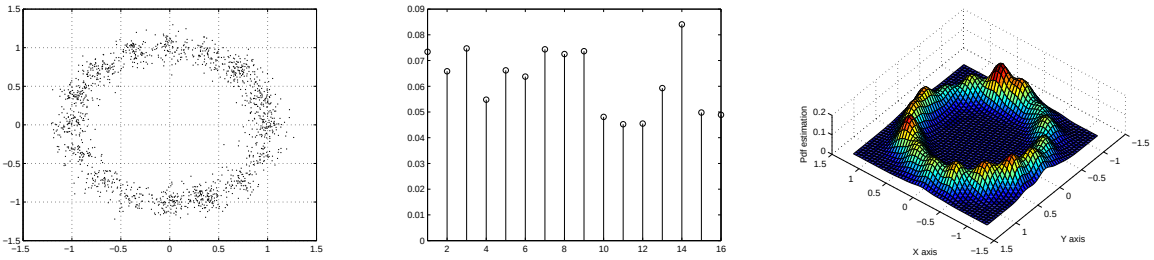
A. Mode Finding

Here, we study the mode finding ability of the two algorithms. We use a systematic approach, by generating a mixture of Gaussian dataset with known modes. We select the kernel size (σ) such that the modes corresponding to the estimated pdf (using Parzen window technique) is as close as possible to the original modes. We then use GMS and GBMS to iteratively track these modes and compare their performance.

1) Dataset 1: Ring of 16 Gaussians with different a priori probabilities (R16Ga): The dataset in Fig. 2(a) consists of a mixture of 16 Gaussians with centers spread uniformly around a circle of unit radius. Each Gaussian density has a spherical covariance of $\sigma_g^2 I = 0.01 \times I$. To include a more realistic scenario, different a priori probabilities were selected which is shown in Fig. 2(b). Using this mixture model, 1500 iid data points were generated. We selected the scale of analysis $\sigma^2 = 0.01$ such that the estimated modes are very close to the modes of the Gaussian mixture. Note that since the dataset is a mixture of 16 Gaussians each with variance $\sigma_g^2 = 0.01$ and spread across the unit circle, the overall variance of the data is much larger than 0.01. Thus by using a kernel size of $\sigma^2 = 0.01$ for Parzen estimation of the pdf, we ensure that the Parzen kernel size is smaller than the actual kernel size of the data. Fig. 2(c) shows the 3D view of this estimated pdf. Note the unequal peaks due to different proportion of points in each cluster.

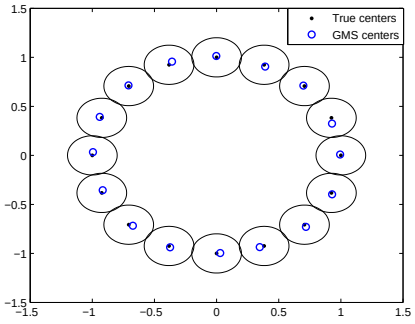
Fig. 3 shows the mode finding ability of the two algorithms. To compare with ground truth we also plot $2\sigma_g$ contour lines and actual centers of the Gaussian mixture. With *tol* level in (17) set to 10^{-6}, GMS algorithm stops at 46^{th} iteration giving almost perfect results. On the other hand, using stopping criterion (18),

[3] See references [2], [3] for more details.

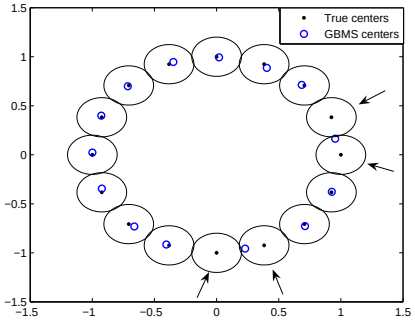


(a) R16Ga Dataset (b) apriori probabilities (c) pdf estimated using $\sigma^2 = 0.01$

Fig. 2. Ring of 16 Gaussian Dataset with different a priori probabilities. The numbering of clusters is in anticlockwise direction starting with center $(1, 0)$



(a) Good Mode finding ability of GMS algorithm



(b) Poor mode finding ability of GBMS algorithm

Fig. 3. Modes of R16Ga Dataset found using GMS and GBMS algorithms

GBMS stops at 20^{th} iteration missing already 4 modes (shown with arrows). We would also like to point out that this is the best result achievable by GBMS even if we had used stopping criterion (17) and selectively hand-picked the best *tol* value.

Fig. 4 shows the cost functions which these algorithms minimize for a duration of 70 iterations. Notice how cost function $H(X)$ of GBMS continuously drops as the modes merge. This would go on until $H(X)$ becomes zero when all the samples would have merged to a single point. For GMS, on the other hand, $H(X; Xo)$ decreases and settles down smoothly as its fixed points (modes) are reached. Thus a more intuitive stopping criterion for GMS which originates directly from its cost function is to stop when the absolute difference between subsequent values of $H(X; Xo)$ became smaller than some *tol* level as summarized below. These are some of the unforeseen advantages when we

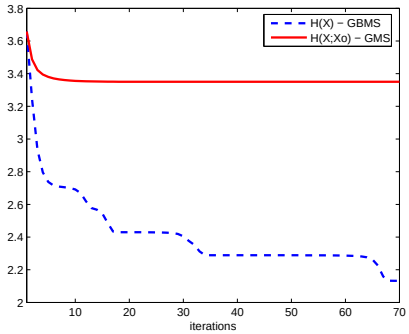


Fig. 4. Cost function of the two algorithms

know exactly what we are optimizing.

$$\left| H(X^{\tau+1}; X_o) - H(X^{\tau}; X_o) \right| < 10^{-10}$$

Another interesting result pops up with this new understanding. Notice that even though GBMS does not directly minimize Renyi's "cross" entropy $H(X; X_o)$, we can always measure this quantity between its result X^{τ} at every iteration τ and the original dataset X_o. If the assumption of two distinct and well separated phases in GBMS holds true, then the samples will quickly collapse to the actual modes of the pdf before they start slowly moving toward each other. Since we start with initialization $X = X_o$, $H(X; X_o)$ will reach its local minimum at this point before it again starts increasing due to the merging of GBMS modes (and hence moving them away from the actual modes of the pdf). By stopping GBMS at this minimum we could devise an effective stopping criterion giving same result as GMS with less number of iterations.

Unfortunately, we found that this works only when the modes (or clusters) are very well separated compared to the kernel size (making the assumption to hold true). For example, Fig. 5 shows $H(X; X_o)$ computed for GBMS for R16Ga dataset. The minimum is reached at 7^{th} iteration. Using this as the stopping criterion would have prematurely stopped GBMS algorithm giving very poor results. It is clear that GBMS is not a good mode finding algorithm.

These results shed a new light in our understanding of these two algorithms. Mode finding can be used as a means to cluster data into different groups. We will see next the performance of these algorithms in clustering where their respective properties effect greatly the outcome of the applications.

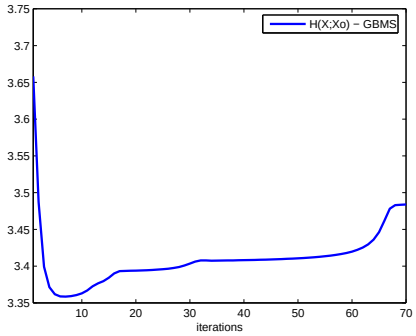


Fig. 5. Renyi's "cross" entropy $H(X; X_o)$ computed for GBMS. This does not work as a good stopping criterion for GBMS in most cases since the assumption of two distinct phases of convergence does not hold true in general.

B. Clustering and Image Segmentation

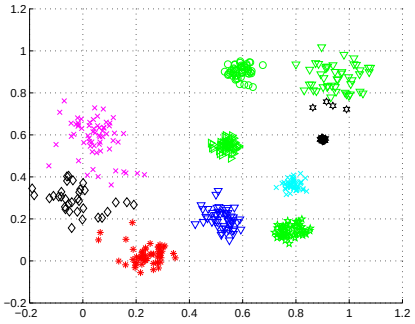
In this section we extend the mode finding ability of GMS to clustering application. We present results on two datasets; the first one is an artificial dataset consisting of different Gaussian clusters and the second one is a real image where we use clustering as a means to segment the image into meaningful objects.

1) Dataset 2: Random Gaussian Clusters (RGC): We generated 10 Gaussian clusters with centers spread uniformly in unit square. The Gaussian clusters have random spherical covariance matrices with 50 iid samples each. Fig. 6 shows the dataset with true labeling as well as the $2\sigma_g$ contour plots.

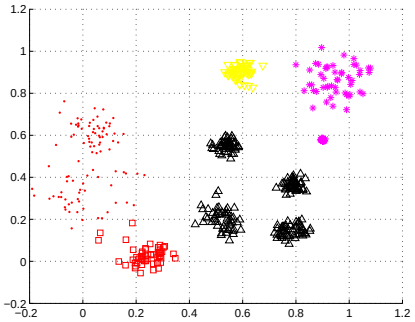
Although, different kernel sizes should be used for density estimation of different clusters, for simplicity and to express our idea clearly we use a common Parzen kernel size for pdf estimation. We found that a $\sigma^2 = 0.01$ performance well for our experiments. The pdf is shown in Fig. 6(c). Note that all the clusters are well identified for this particular kernel size. By correlating the points with their respective modes we wish to segment this dataset into meaningful clusters.

With *tol* level set at 10^{-6} the GMS algorithm converges at 41^{st} iteration. The segmentation result is shown in Fig. 7(a) . Clearly GMS performs very well in clustering the dataset into meaningful clusters. There are a total of 20 misclassification (out of 500 points) which arise mostly due to the cluster with the largest spherical covariance matrix. Notice that this cluster is underrepresented with just 50 points. Further due to the overlap of the $2\sigma_g$ contour of this cluster with the neighboring cluster as shown in Fig. 6(b), the misclassifications are bound to occur. Another interesting mistake occur at the top right corner, where 4 points belonging to a cluster are misclassified and put as part of another highly concentrated cluster. These points lie in the narrow valley bordering the two clusters and unfortunately their gradient directions point toward the incorrect mode. But it should be appreciated that even for this complex dataset with varying shapes of Gaussian clusters, GMS with the simplest solution of single kernel size gives such a good result.

On the other hand, using stopping criterion (18), GBMS stops at 18^{th} with the output shown in Fig. 7(b). Notice the poor segmentation result as a consequence of multiple modes merging. It should be kept in mind that by defining the kernel size $\sigma^2 = 0.01$, we have selected the similarity measure for clustering and are looking for spherical Gaussians with variance around this value. In this regard, the result of GBMS is incoherent. On the other hand, the segmentation result obtained for GMS is much more homogeneous and consistent with our similarity measure. Further, it is only in case of GMS that the modes estimated from the pdf directly translate into clusters. On the contrary, for GBMS its not clear how the modes in Fig. 6(c) correlate with the clustering solution obtained in Fig. 7(b).



(a) Segmentation result using GMS



(b) Segmentation result using GBMS

Fig. 7. Segmentation results of RGC dataset using the two algorithms

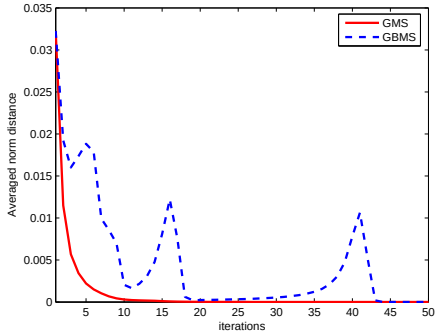
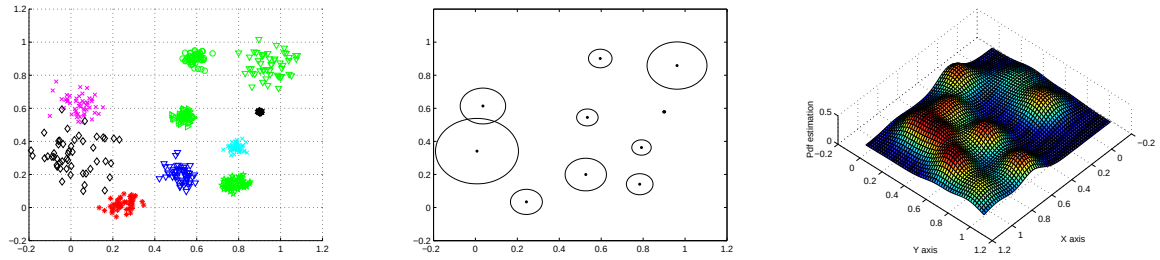


Fig. 8. Averaged Norm Distance moved by particles in each iteration

Fig. 8 shows the average change in particle position for both the algorithms. Notice the peaks in GBMS curve corresponding to modes merging. This is a classic example were the assumption of two distinct phases in GBMS becomes fuzzy. By 5^{th} iteration, two of the modes have already merged and by 18^{th} iteration a total of 5 modes are lost giving rise to poor segmentation result. In case of GMS, on the other hand, the averaged norm distance steadily decreases and by selecting a *tol* level sufficiently low, we are always assured a good segmentation result.



(a) RGC Dataset (b) $2\sigma_g$ contour plots (c) pdf estimated using $\sigma^2 = 0.01$

Fig. 6. Random Gaussian Clusters Dataset, its $2\sigma_g$ contour plots and its estimated pdf

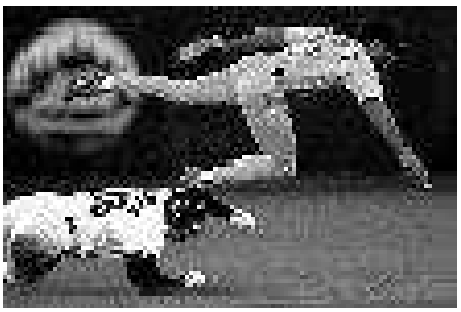


Fig. 9. Baseball Image

2) Dataset3: Baseball Game Image: We highlight the differences between GMS and GBMS by applying it on a real dataset. For this purpose, we use the famous baseball game image of the normalized cuts paper by Shi and Malik [16] shown in Fig. 9. For computation purpose, the image has been reduced to 110×73 pixels. This gray level image is transformed to 3 dimensional feature space consisting of two spatial features namely the x, y coordinates of the pixels and the range feature which is the intensity value at that location. Thus the dataset consists of 8030 points in the feature space. In order to use an isotropic kernel we prescale the intensity value such that they fall in the same range as the spatial features as done in [4]. All the values reported are in pixel units.

In images, we found that a more efficient stopping criterion for GMS is to stop when the maximum distance moved among all the particles is less than some *tol* level rather than the average distance. This is summarized in (19). Further, we set the *tol* level equal to 10^{-3} for both the algorithms throughout this experiment.

$$\text{Stop when} \quad \max_i \|x_i^{(\tau)} - x_i^{(\tau-1)}\| < tol \tag{19}$$

We performed an elaborate experiment of multi scale analysis where the kernel size σ was changed from a small value to large value in steps of 0.5. We selected the best segmentation result for both the algorithms for a particular number of segments. The results are shown in Fig. 10. The first column shows the segmentation result for 8 clusters. Since the clusters are well separated for the respective kernel sizes, both GMS and GBMS give very similar results. The interesting development occurs when we try to achieve segments less than 8. Note that for this image the best number of segments is 5 to 6 segments as seen in the image itself. Many researcher have tried to do this using various methods [4], [16].

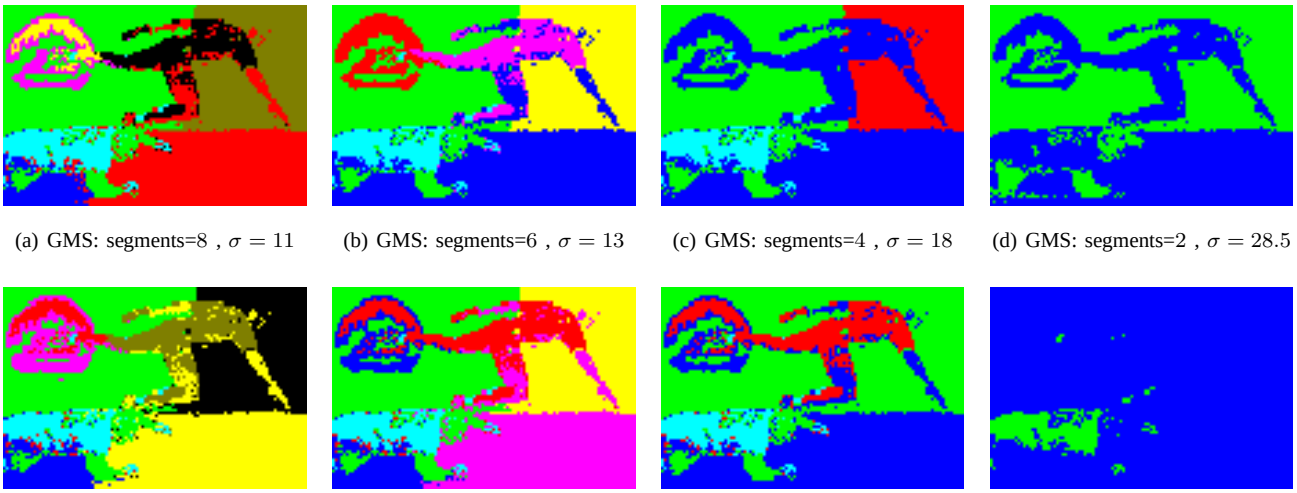
Fig. 10(b) and Fig. 10(f) shows the GMS and GBMS result for 6 segments. Note the poor performance of GBMS. Instead of grouping similar objects into one, GBMS splits them and merges half to two different clusters. The disc segment in the image was split into two with one of them merging with the player and the other with the bottom background. This is counter intuitive given the fact that two of the coordinates of the feature space are spatial coordinates of the image. On the other hand, GMS clearly gives a very good segmentation result with each segment corresponding to an object in the image. Further, a nice consistent and hierarchical structure is seen in GMS. As we reduce the number of clusters, GMS merges clusters of same intensity and which are closer to each other before merging similar intensity clusters which are far apart. This is what we would expect for this feature space. This results in a beautiful pattern in the image space where whole objects which are similar are merged together in an intuitive manner. This phenomenon is again observed as we move from 6 segments to 4 where GMS puts all the gray objects in one cluster thus putting together three full objects of similar intensity in one group.

Thus starting from 8 segments result which were very similar to each other, GMS and GBMS tread a very different path for lower number of segments. GMS neatly segments objects in the image into different segments and hence is very close to human segmentation result. The different path followed by the two algorithms results in a completely different 2 level image segmentation as shown in Fig. 10.

V. Discussion and Conclusions

Mean shift, a mode seeking technique, has become increasingly popular in image processing and vision community to perform clustering, segmentation and tracking. Two competing algorithm are GMS and GBMS, which differ slightly in the way the fixed point equation is updated. In this paper, we have successfully analyzed these algorithms from an optimization framework using information theoretic concepts. To the best of our knowledge, this is the first such comprehensive study of these two algorithms after Cheng's work [2].

With this new understanding a number of interesting results follow. We have shown that GBMS directly minimizes Renyi's quadratic entropy and hence is an unstable mode finding algorithm. Since modes are neither stationary nor saddle points of this cost function, any stopping criterion would at most be heuristic. On the other hand, its stable counterpart GMS, minimizes Renyi's "cross" entropy reaching its local minimum when the modes are reached. Thus a new stopping criterion is to stop when the change



(a) GMS: segments=8 , $\sigma = 11$ (b) GMS: segments=6 , $\sigma = 13$ (c) GMS: segments=4 , $\sigma = 18$ (d) GMS: segments=2 , $\sigma = 28.5$

(e) GBMS: segments=8 , $\sigma = 10$ (f) GBMS: segments=6 , $\sigma = 11.5$ (g) GBMS: segments=4 , $\sigma = 13$ (h) GBMS: segments=2 , $\sigma = 18$

Fig. 10. Baseball image segmentation using GMS and GBMS algorithms. The top row shows results from GMS for various different number of segments and the σ at which it was achieved. The bottow row similarly shows the results from GBMS

in the cost function is small. Through extensive experiments we have shown how this new perspective effects greatly the outcome of these two algorithms.

This idea can also be extended to mean shift with any other kernel. A pdf estimated with kernel K_1 will result in entropy estimator with kernel K_2, with K_2 being a convolution of K_1 with itself. Differentiating this estimator would give us fixed point update with kernel K_3. Thus, mean shift with K_3 would result in gradient ascent on density estimated with kernel K_2 which was given a special name called "shadow" kernel in [2]. We could give a similar name like "preshadow" kernel to K_1 which is only needed to complete the theory but never used in practice.

Another issue we haven't addressed here is kernel density estimation which in itself is a vast and well researched field. We would direct the readers to [3] for more details on this topic. However, an important point needs special mention at this stage. As an example take the RGC dataset. Proper density estimation would assign larger kernel size to samples of broad clusters and smaller to samples of compact clusters. This would only improve the density estimation giving even better results for the GMS. On the other hand, due to dramatically different rates at which these clusters collapse to their modes, stopping GBMS would become even harder giving poor results.

To conclude, we hope that our new insight would foster fresh interest in this exciting field and pave the way for even better understanding of these mean shift algorithms.

Acknowledgment

This work was partially supported by NSF grant ECS-0601271 and ECS-0422718.

References

[1] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition," *IEEE Trans. on Information theory*, vol. 21, no. 1, pp. 32–40, January 1975.

[2] Y. Cheng, "Mean shift, mode seeking and clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, August 1995.

[3] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[4] M. Carreira-Perpiñán, "Gaussian mean shift is an EM algorithm," *To appear in IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2007.

[5] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, June 2003, pp. 467–474.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, June 2000, pp. 142–149.

[7] M. Carreira-Perpiñán, "Mode-finding for mixtures of gaussian distributions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1318–1323, November 2000.

[8] M. Á. Carreira-Perpiñán, "Fast nonparametric clustering with gaussian blurring mean-shift." in *ICML*, W. W. Cohen and A. Moore, Eds. ACM, 2006, pp. 153–160.

[9] M. Fashing and C. Tomasi, "Mean shift is a bound optimization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 471–474, March 2005.

[10] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[11] A. Renyi, "On measure of entropy and information," in *Proceedings 4th Berkeley Symp. Math. Stat. and Prob.*, vol. 1, 1961, pp. 547–561.

[12] J. C. Principe, D. Xu, and J. Fisher, "Information theoretic learning," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. John Wiley, 2000, pp. 265–319.

[13] D. Erdogmus, "Information theoretic learning: Renyi's entropy and its applications to adaptive system training," Ph.D. dissertation, University of Florida, 2002.

[14] R. Jenssen, "An information theoretic approach to machine learning," Ph.D. dissertation, University of Tromso, 2005.

[15] S. Rao, W. Liu, J. C. Principe, and A. de Medeiros Martins, "Information theoretic mean shift algorithm," in *Proceedings of IEEE Conf. on Machine Learning for Signal Processing*, Sept. 2006, pp. 155–160.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.