Stochastic Gradient Kernel Density Mode-Seeking

Xiao-Tong Yuan
NLPR, CASIA
xtyuan@nlpr.ia.ac.cn

Stan Z. Li
CBSR&NLPR, CASIA
szli@nlpr.ia.ac.cn

Abstract

As a well known fixed-point iteration algorithm for kernel density mode-seeking, Mean-Shift has attracted wide attention in pattern recognition field. To date, Mean-Shift algorithm is typically implemented in a batch way with the entire data set known at once. In this paper, based on stochastic gradient optimization technique, we present the stochastic gradient Mean-Shift (SG-MS) along with its approximation performance analysis. We apply SG-MS to the speedup of Gaussian blurring Mean-Shift (GBMS) clustering. Experiments in toy problems and image segmentation show that, while the clustering accuracy is comparable between SG-GBMS and Naive-GBMS, the former significantly outperforms the latter in running time.

1. Introduction

In many computer vision applications, e.g., video tracking [9] and image segmentation [8], it is necessary to design algorithms to find the clusters of a data set sampled from some unknown distribution. Probability density estimation (PDE) may represent the distribution of data in a given problem and then the modes can be taken as the representatives of clusters. A family of non-parametric PDE, namely kernel density estimation (KDE) [13], is mostly applied in practice for its ability to describe the potential distribution in a data-driven way. Given a data set $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$ drawn from a population with density function $f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d$, the general multi-variable KDE with kernel $k(\cdot)$ is defined by

$$\hat{f}_k(\mathbf{x}) = \frac{1}{NC_k} \sum_{n=1}^{N} k(M^2(\mathbf{x}, \mathbf{x}_n, \mathbf{\Sigma})) \qquad (1)$$

where $M^2(\mathbf{x}, \mathbf{x}_n, \mathbf{\Sigma}) = (\mathbf{x} - \mathbf{x}_n)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mathbf{x}_n)$ is the Mahalanobis distance from $\mathbf{x}$ to $\mathbf{x}_n$ with bandwidth matrix $\mathbf{\Sigma}$ and C_k is a normalization constant.

To find the local maximum from a starting point $\mathbf{y}_0$, the Mean-Shift (MS) optimization algorithm solves the gradient equation of (1) via the following fixed point iteration

$$\mathbf{y}_m = \frac{\sum_{n=1}^N g(M^2(\mathbf{y}_{m-1}, \mathbf{x}_n, \mathbf{\Sigma}))\mathbf{x}_n}{\sum_{n=1}^N g(M^2(\mathbf{y}_{m-1}, \mathbf{x}_n, \mathbf{\Sigma}))} \qquad (2)$$

where profile $g(\cdot) = -k^{'}(\cdot)$. The MS algorithm can be applied to data clustering by declaring each mode of the KDE as representative of one cluster, and assigning each data point $\mathbf{x}_n$ to the mode it converges to via iteration (2). Since the algorithm does not depend on parameters such as step sizes, the clustering is uniquely defined given the KDE, i.e., given the bandwidth matrix $\mathbf{\Sigma}$. Also, different kernels give rise to different versions of the MS algorithm [8]. More numerical analysis and extensions of MS can be found in [6][7][8][10][15].

To date, the MS density mode-seeking is carried out in a batch way on the entire training set. The motivation of this work is to effectively implement MS in an "incremental" or "online" manner for local KDE mode-seeking. Generally speaking, there are two reasons why incremental methods may be useful. First, a learning or optimization algorithm has a competitive advantage if it can immediately use all the training data collected so far, rather than wait for a complete training set. Second, incremental algorithms usually have smaller memory requirements: new training examples are used to update a "state" and then the examples are forgotten. The state summarizes the information collected so far - it typically consists of a parametric model and it thus occupies a much smaller amount of memory than a full-fledged training set. Specially, an online KDE mode-seeking algorithm should be able to update a local KDE mode on each data instance arrival by maintaining hypothesis that reflects all the data instances seen so far.

The online KDE mode-seeking algorithm developed in this paper is a stochastic gradient ascent variation of MS. It is naturally motivated by the knowledge that MS is essentially an adaptive step-size gradient ascent method [7]. We show that with a slight modification of batch iteration (2), MS can be extended into a stochastic gradient ascent algorithm with time variable learning rate. The almost sure convergence of our stochastic gradient MS is analyzed under some preliminaries and the convergence speed is shown to

be generally sub-linear. We also derive the regret bound of our algorithm by viewing it as an online programming [17]. Furthermore, we apply our algorithm to the acceleration of the popularly used Gaussian blurring MS (GBMS) clustering method. This is done by moving the data points at each GBMS iteration according to stochastic gradient MS instead of batch MS, which leads to 2-3 times speedup of convergence for GBMS in image segmentation tasks.

The remainder of this paper is organized as follows: Section 2 presents the algorithm development and convergence analysis for stochastic gradient MS. Section 3 provides an application of our algorithm to GBMS acceleration. We conclude this work in section 4.

2. Stochastic Gradient Mean-Shift

Our stochastic gradient MS (SG-MS) is a natural online extension of the iteration process (2). It is a stochastic gradient ascent method with adaptive learning rate for KDE (1). We first present this algorithm and then give some main results on its asymptotical properties.

2.1. Algorithm

The formal description of SG-MS algorithm (as a function pseudocode) is given in algorithm 1. The key point is to maintain a vector accumulator $\mathbf{R}$ and a scalar accumulator S that correspond to the numerator and denominator in iteration (2) respectively. The algorithm works as follows: At arrival of each sample $\mathbf{x}_n$, its weighted version $g(M^2(\hat{\mathbf{y}}_{n-1}^o, \mathbf{x}_n, \mathbf{\Sigma}))\mathbf{x}_n$ and the weight $g(M^2(\hat{\mathbf{y}}_{n-1}^o, \mathbf{x}_n, \mathbf{\Sigma}))$ are integrated into $\mathbf{R}$ and S respectively. The local mode $\hat{\mathbf{y}}_n^o$ is then updated as the quotient of $\mathbf{R}$ and S. Note that the closer $\mathbf{x}_n$ is to $\hat{\mathbf{y}}_{n-1}^o$, the more it will contribute to the output $\hat{\mathbf{y}}_n^o$.

SG-MS(Query point $\mathbf{y}_0$, Data set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, Bandwidth $\mathbf{\Sigma}$) {***Function interface***}
Let $\hat{\mathbf{y}}_0^o = \mathbf{y}_0$, initialize $\mathbf{R} \leftarrow 0$, $S \leftarrow 0$.
for $n = 1, ..., N$ **do** {***Online Update***}
  $\mathbf{R} \leftarrow \mathbf{R} + g(M^2(\hat{\mathbf{y}}_{n-1}^o, \mathbf{x}_n, \mathbf{\Sigma}))\mathbf{x}_n$
  $S \leftarrow S + g(M^2(\hat{\mathbf{y}}_{n-1}^o, \mathbf{x}_n, \mathbf{\Sigma}))$
  $\hat{\mathbf{y}}_n^o \leftarrow \frac{\mathbf{R}}{S}$
end for
Return $\hat{\mathbf{y}}_N^o$.

Algorithm 1: Stochastic Gradient Mean-Shift

2.2. Relation to Stochastic Gradient Optimization

It is easy to see that SG-MS can be written in an iterative form as

$$\hat{\mathbf{y}}_n^o = \frac{\sum_{s=1}^n g(M^2(\hat{\mathbf{y}}_{s-1}^o, \mathbf{x}_s, \mathbf{\Sigma}))\mathbf{x}_s}{\sum_{s=1}^n g(M^2(\hat{\mathbf{y}}_{s-1}^o, \mathbf{x}_s, \mathbf{\Sigma}))}.$$

To simplify the notations, we denote $H(\mathbf{x}, \mathbf{y}) = \nabla_{\mathbf{y}} k(M^2(\mathbf{y}, \mathbf{x}, \mathbf{\Sigma}))$. The above iteration equation is equivalent to

$$\hat{\mathbf{y}}_n^o = \hat{\mathbf{y}}_{n-1}^o + \eta_n \mathbf{\Sigma} H(\mathbf{x}_n, \hat{\mathbf{y}}_{n-1}^o) \tag{3}$$

where $\eta_n = \left(2\sum_{s=1}^n g(M^2(\hat{\mathbf{y}}_{s-1}^o, \mathbf{x}_s, \mathbf{\Sigma}))\right)^{-1}$. The iteration (3) is a stochastic gradient ascent optimization [4][11] for KDE (1) with time dependent learning rate $\eta_n \mathbf{\Sigma}$.

2.3. Approximation Analysis

Bottou and Cun [4] pointed out that approximation of stochastic gradient ascent towards local maximum of *empirical cost* (e.g., KDE (1) in this work) on a finite data set is hopelessly slow. Instead, they recommended concentrating on the convergence towards local maximum of the *expected cost*. We follow this suggestion by considering the situation where the supply of input data samples is essentially unlimited ($N \to \infty$) and randomly drawn from some unknown density $f(\mathbf{x})$. We will focus on the approximation of SG-MS towards the local maximum $\mathbf{y}^*$ of the following *expected kernel density estimation* (EKDE)

$$f_k(\mathbf{y}) \triangleq \frac{1}{C_k} \int k(M^2(\mathbf{y}, \mathbf{x}, \mathbf{\Sigma}))f(\mathbf{x})d\mathbf{x}.$$

2.3.1 Preliminaries

The convergence results rely on the following assumptions:

- The EKDE function $f_k(\mathbf{y})$ is bounded above and three times differential with continuous derivatives.
- In the domain of interest, the profile function $g(\cdot)$ is positively bounded below and above.

$$\exists L_1, L_2, \quad L_1 < g(\cdot) < L_2 \tag{4}$$

- The second moment of the update term $H(\mathbf{x}, \mathbf{y})$ should grow more than quadratic w.r.t. the norm of $\mathbf{y} - \mathbf{y}^*$

$$E_{\mathbf{x}}(\|H(\mathbf{x}, \mathbf{y})\|^2) \leq A + B\|\mathbf{y} - \mathbf{y}^*\|^2. \tag{5}$$

- When the norm of $\mathbf{y} - \mathbf{y}^*$ is larger than a certain horizon D, the gradient $\nabla_{\mathbf{y}} f_k(\mathbf{y})$ points towards the $\mathbf{y}^*$

$$\sup_{\|\mathbf{y}-\mathbf{y}^*\|^2 > D} (\mathbf{y} - \mathbf{y}^*)^T \nabla_{\mathbf{y}} f_k(\mathbf{y}) < 0 \tag{6}$$

- When norm of $\mathbf{y} - \mathbf{y}^*$ is smaller than a second horizon E greater than D, the norm of the update term $H(\mathbf{x}, \mathbf{y})$ is bounded regardless of $\mathbf{x}$.

$$\forall \mathbf{x}, \quad \sup_{\|\mathbf{y}-\mathbf{y}^*\|^2 < E} \|H(\mathbf{x}, \mathbf{y})\| \leq K_0 \tag{7}$$

2.3.2 Almost Sure Convergence

First, we introduce the following straightforward lemma on learning rate η_n:

Lemma 1 *Given the assumption (4), we have that*

$$\sum_{n=1}^{\infty} \eta_n = \infty \quad \textit{and} \quad \sum_{n=1}^{\infty} \eta_n^2 < \infty.$$

Based on lemma 1 and the *gradient convergence theorem* in [3], we obtain the following convergence result:

Proposition 1 *Given the assumptions listed in Section 2.3.1, the random gradient sequence* $\{\nabla_{\mathbf{y}} f_k(\hat{\mathbf{y}}_n^o)\}_{n=1,2,...}$ *generated by SG-MS converges almost sure to zero.*

This proposition implies that SG-MS will soon or later reach the *final convergence phase* [2].

2.3.3 On Convergent Speed

We further discuss the convergent speed of SG-MS towards $\mathbf{y}^*$. Denote the *ensemble mean* [11] at sample time stamp n by $\mathbf{y}_n^o = \mathscr{E}(\hat{\mathbf{y}}_n^o)$, and let $\mathcal{Q} = \mathbf{E}_{\mathbf{x}}\left(\nabla_{\mathbf{y}} H(\mathbf{x}, \mathbf{y}^*)\right)$. Here $\mathscr{E}$ denotes the mean under all the possible sequences on $\mathcal{X}$. The following proposition is a straightforward result by [11, Lemma 6].

Proposition 2 *If the estimated mode* $\hat{\mathbf{y}}_n^o$ *is in the vicinity of* $\mathbf{y}^*$*, the evolution of ensemble mean is approximated by the following recursive equation:*

$$\mathbf{y}_n^o = \mathbf{y}_{n-1}^o + \eta_n \boldsymbol{\Sigma} \mathcal{Q} (\mathbf{y}_{n-1}^o - \mathbf{y}^*).$$

From proposition 2 we know that $\mathbf{y}_n^o - \mathbf{y}^* = (\mathbf{I} + \eta_n \boldsymbol{\Sigma}\mathcal{Q})(\mathbf{y}_{n-1}^o - \mathbf{y}^*)$. Given the assumption (4), we have that $\eta_n \to 0$ as $n \to \infty$. Therefore $\lim\limits_{n\to\infty} \frac{\|\mathbf{y}_n^o - \mathbf{y}^*\|}{\|\mathbf{y}_{n-1}^o - \mathbf{y}^*\|} = 1$, which indicates that the convergent speed of $\mathbf{y}_n^o$ towards $\mathbf{y}^*$ is generally sub-linear [12].

It is interesting to further consider the case that kernel function $k(\cdot)$ is piecewise linear, in which the traditional batch MS is equivalent to Newton-Raphson optimization [10]. Since $\mathbf{y}_n^o \to \mathbf{y}^*$, it can be known through law of large numbers that $\eta_n \boldsymbol{\Sigma} \to -\frac{1}{n}\mathcal{Q}^{-1}$ in this case. Therefore, according to [4, Theorem 3], we may derive the following result on L^2-norm convergent speed of SG-MS:

Proposition 3 *When kernel function* $k(\cdot)$ *is piecewise linear, we have that*

$$\mathbf{E}\left(\|\hat{\mathbf{y}}_n^o - \mathbf{y}^*\|^2\right) = \frac{\mathcal{Q}^{-1}\mathcal{G}\mathcal{Q}^{-1}}{n} + \mathcal{O}(\frac{1}{n^2})$$

where $\mathcal{G} = \mathbf{E}_{\mathbf{x}}(H(\mathbf{x}, \mathbf{y}^*)H(\mathbf{x}, \mathbf{y}^*)^T)$.

2.3.4 On Regret Bound

For online gradient ascent (or descent) optimization, the difference between the total profit (or loss) and its optimal value for an off-line fixed action is known as the *regret* [1][17]. Here, in the context of online KDE maximization problem, we may take the mode $\hat{\mathbf{y}}_{n-1}^o$ as the action at time stamp n, and $k(M^2(\hat{\mathbf{y}}_{n-1}^o, \mathbf{x}_n, \boldsymbol{\Sigma}))$ as the current profit. Denote $k_n(\mathbf{x}) = k(M^2(\mathbf{x}, \mathbf{x}_n, \boldsymbol{\Sigma}))$. Then the regret of SG-MS is calculated as

$$\mathcal{R}_N = \frac{1}{NC_k}\left[\sum_{n=1}^{N} k_n(\hat{\mathbf{y}}_{n-1}^o) - \max_{\mathbf{y}}\left(\sum_{n=1}^{N} k_n(\mathbf{y})\right)\right].$$

The following proposition gives a lower bound of this regret:

Proposition 4 *Suppose that* k *is concave and* $\exists G > 0$ *so that* $M^2(k_n^{'}(\mathbf{x}), \mathbf{0}, \boldsymbol{\Sigma}^{-1}) \le G^2$*, then the regret of SG-MS is bounded below as*

$$\mathcal{R}_N \ge -\frac{1}{2NC_k}\sum_{n=1}^{N} \eta_n G^2.$$

Given condition (4), we may loose the bound as

$$\mathcal{R}_N \ge -\frac{G^2}{4L_1 C_k}\left(\frac{1+\log N}{N}\right).$$

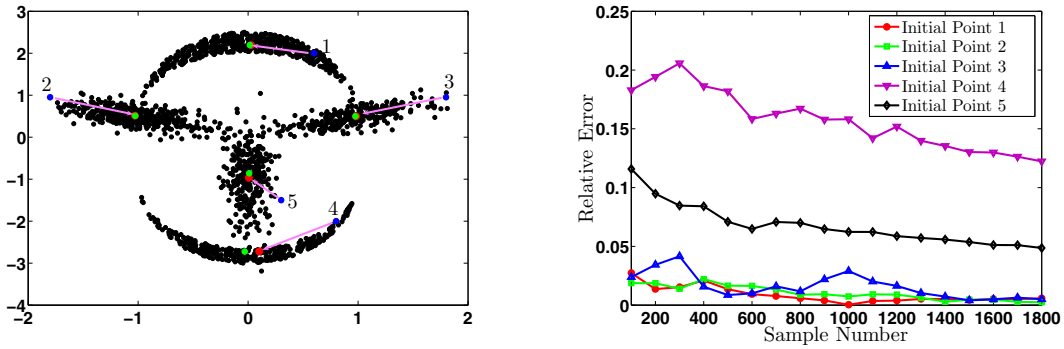
Therefore $\lim_{N\to\infty} \mathcal{R}_N \ge 0$ *holds.*

2.4. A Numerical Example

As a numerical test, we employ SG-MS algorithm to perform local KDE mode-seeking on a 2D toy data set (shaped as shown in Figure 1(a), size $1,821$). For computational simplicity, the isotropic covariance $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ is used throughout the rest of this paper. We start from five initial points (represented by blue dots in Figure 1(a)) to locate the corresponding local modes using SG-MS and batch MS separately. The modes located by SG-MS after one pass of scanning of the whole data set are shown in red dots in Figure 1(a), while the green ones represent those modes returned by batch MS. For each initial point $\mathbf{y}_0$, the relative error between the returned modes from it by SG-MS ($\hat{\mathbf{y}}_*^o$) and batch MS ($\hat{\mathbf{y}}_*^b$) is calculated as $\|\hat{\mathbf{y}}_*^o - \hat{\mathbf{y}}_*^b\|/\|\hat{\mathbf{y}}_*^b\|$. We plot the relative error evolving curves of SG-MS from the five initial points in Figure 1(b). From these curves we can see that SG-MS generally makes satisfying numerical approximation towards batch MS.

3. Application to GBMS Speedup

To show the practical usage of our SG-MS algorithm, we apply it to the fast implementation of the Gaussian blurring



(a) (b)

Figure 1. Numerical test of SG-MS on a 2D toy data set. Bandwidth $\sigma = 0.5$. (a) Detected Modes by SG-MS (in red) and batch MS (in green) from five different initial points (in blue). (b) The relative error evolving curves.

MS (GBMS) clustering method, with performance evaluation on image segmentation tasks.

The GBMS [5][7] clustering method uses Gaussian kernel in KDE and iteratively sharpens the data set by moving each data point according to MS. Algorithm 2 formally presents the GBMS clustering method, which we refer to as Naive-GBMS in the following context. Perpinan [5] proves that Naive-GBMS converges cubically and thus is much faster than traditional Gaussian MS [8] in which the data set is fixed during MS calculation.

As can be seen from algorithm 2 that, at each iteration, each point moves to the output returned by one step of batch MS iteration. In this way the data set evolves and quickly collapses into clusters. We claim that such a data set evolving mechanism can be improved by using our SG-MS to more quickly sharpen the data set. This is comprehensive since SG-MS typically shifts a point *close* to its corresponding local mode in just one pass of data set visiting, given that data set is abundant and thoroughly sampled. A formal description of our stochastic gradient GBMS (SG-GBMS) method is given in algorithm 3.

Let initial data set be $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$.
while Convergence is not attained **do**
  for each $\mathbf{x}_m \in \mathcal{X}$ **do**

$$\mathbf{y}_m = \frac{\sum_{n=1}^N e^{-\frac{\|\mathbf{x}_m - \mathbf{x}_n\|^2}{\sigma^2}} \mathbf{x}_n}{\sum_{n=1}^N e^{-\frac{\|\mathbf{x}_m - \mathbf{x}_n\|^2}{\sigma^2}}}$$

  end for
  $\mathcal{X} \leftarrow \{\mathbf{y}_m\}_{m=1}^N$
end while
Clustering: connected_components($\mathcal{X}$, min_diff).

Algorithm 2: Gaussian Blurring Mean-Shift

We still use the toy 2D data set mentioned in 2.4 to illustrate the numerical performance of SG-GBMS. The data set evolving and final clustering results of SG-GBMS and

Let initial data set be $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$.
while Convergence is not attained **do**
  for each $\mathbf{x}_m \in \mathcal{X}$ **do**
    $\mathbf{y}_m = \text{SG-MS}(\mathbf{x}_m, \mathcal{X}, \sigma^2\mathbf{I})$
  end for
  $\mathcal{X} \leftarrow \{\mathbf{y}_m\}_{m=1}^N$
end while
Clustering: connected_components($\mathcal{X}$, min_diff).

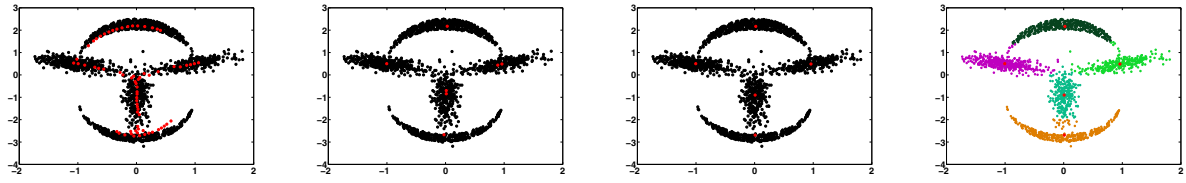
Algorithm 3: Stochastic Gradient GBMS

Naive-GBMS are given in Figure 2. As can be seen from Figure 2(a)∼2(c) and Figure 2(e)∼2(g) that SG-GBMS sharpens data set faster than Naive-GBMS does. The necessary iteration times before convergence are three and seven for SG-GBMS and Naive-GBMS respectively. The final clustering results of these two algorithms are given in Figure 2(d)&2(h), which are visually comparable. We quantitatively evaluate the overall approximation performance of SG-GBMS towards Naive-GBMS by using the following defined *ε-error rate*(ε-ER)[16] :

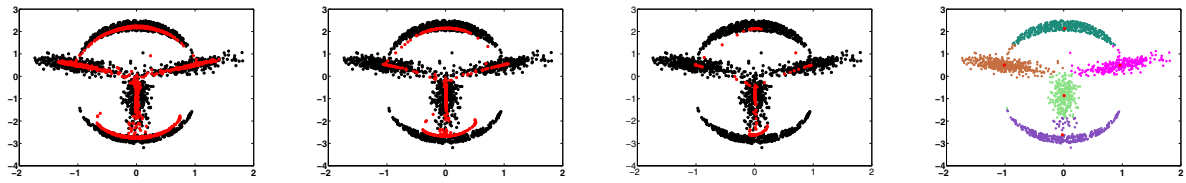
$$\varepsilon\text{-ER} = \frac{1}{N}\sum_{n=1}^{N} \delta\left(\frac{\|\hat{\mathbf{y}}_{n,*}^o - \hat{\mathbf{y}}_{n,*}^b\|}{\|\hat{\mathbf{y}}_{n,*}^b\|} > \varepsilon\right)$$

where $\delta(x)$ is the delta function that equals one if boolean variable x is true and equals zero otherwise, while $\hat{\mathbf{y}}_{n,*}^o$ and $\hat{\mathbf{y}}_{n,*}^b$ are convergent modes returned by SG-GBMS and Naive-GBMS respectively from an initial query point $\mathbf{x}_n$ in $\mathcal{X}$. We set $\varepsilon = 0.05$ in this case, and as a result the ε-ER achieved by SG-GBMS is 0.0033.

One important application of GBMS clustering algorithm is for unsupervised image segmentation. We test in this part the performance of SG-GBMS in large size image segmentation tasks. We follow the approaches in [8][14], where each datum is represented by *spatial-range* joint features (i, j, r, g, b), where (i, j) is the pixel location in the



(a) SG-GBMS, iteration=1 (b) SG-GBMS, iteration=2 (c) SG-GBMS, iteration=3 (d) SG-GBMS, Clusters



(e) Naive-GBMS, iteration=1 (f) Naive-GBMS, iteration=2 (g) Naive-GBMS, iteration=3 (h) Naive-GBMS, Clusters

Figure 2. The data set evolving and clustering results of SG-GBMS and Naive-GBMS. The black points represent the original data set while the red points represent the currently updated data set. We set bandwidth $\sigma = 0.5$. (a)$\sim$(c): The data set evolving results of SG-GBMS after one, two and three times of iteration respectively. (d) The converged clustering result of SG-GBMS after 3 times of iteration. (e)$\sim$(g): The data set evolving results of Naive-GBMS after one, two and three times of iteration respectively. (h) The converged clustering result of Naive-GBMS after seven times of iteration.

image and (r, g, b) is the normalized RGB color feature. Figure 4 shows the results by SG-GBMS and Naive-GBMS on the color image *hand* (137×110) under three different kernel bandwidths. The quantitative evaluation curves are plotted in Figure 3, from which we can clearly see the speedup advantage of SG-GBMS over Naive-GBMS with bounded approximation error.

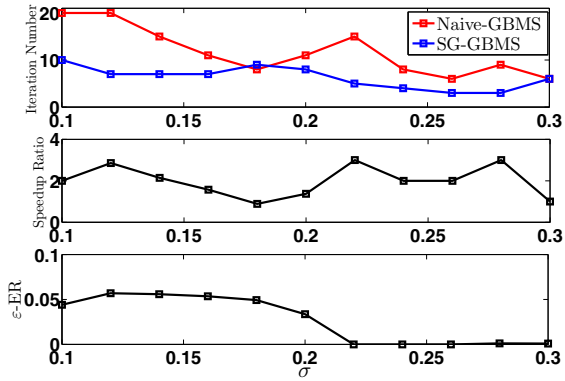


Figure 3. Quantitative evaluation curves for the *hand* image. Top: Iteration Number *vs.* bandwidth; Middle: Speedup Ratio *vs.* bandwidth; Bottom: ε-ER ($\varepsilon = 0.1$) *vs.* bandwidth.

Some images from the Berkeley segmentation data set [1] are also used for evaluation. Four selected groups of segmentation results are given in Figure 5. The quantitative comparison between SG-GBMS and Naive-GBMS on these images are listed in Table 1, from which we can see that SG-GBMS converges faster than Naive-GBMS does. The

[1] http://www.eecs.berkeley.edu/Research/ Projects/CS/vision/bsds/

(a) (b) (c) (d) (e) (f)

Figure 4. Segmentation image pairs of SG-GBMS (left) and Naive-GBMS (right) on the *hand* image under bandwidth $\sigma = 0.1$, 0.2 and 0.3 separately.

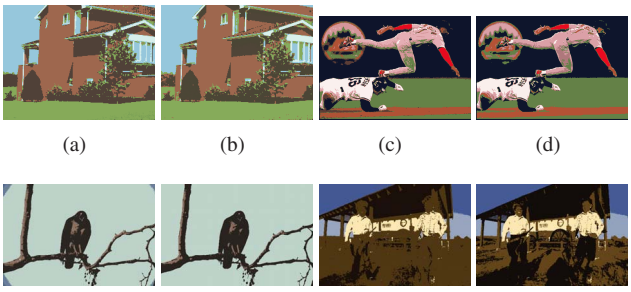
ε-ER introduced by SG-GBMS on the first three images are acceptable and the segmentation results are visually comparable to those by Naive-GBMS. However, SG-GBMS performs poorly in clustering accuracy for the *cowboy* image, as can be seen from the last column of Table 1 and Figure 5(g)&5(h). Our SG-GBMS fails to discriminate the black and brown color regions. This failing case reminds us that sometimes an incremental method will not be as efficient as a batch method in data information extraction. This is because decision must often be made without the benefit of future information.

4. Conclusion and Future Work

Based on the technique of stochastic gradient optimization, we present in this paper the SG-MS algorithm for incremental KDE mode-seeking. Theoretically, we have shown that SG-MS converges in sub-linear speed and asymptotically there is no regret with respect to batch MS. Numerical tests validate that our SG-MS typically performs comparably to its batch mode counterpart, given that enough data samples are available. We have applied SG-MS to the algorithm speedup of GBMS. Experiments in image segmentation show that, while the clustering accuracy of

Table 1. Quantitative results by SG-GBMS and Naive-GBMS on four test images. ($\varepsilon = 0.1$)

Images		*House*	*Base Dive*	*Hawk*	*Cowboy*
Sizes		255×192	432×294	481×321	481×321
σ		0.1	0.1	0.06	0.1
Iteration Number	SG-GBMS	9	7	22	12
	Naive-GBMS	20	27	58	29
Speedup Ratio		2.22	3.86	2.64	2.42
ε-ER		0.053	0.095	0.035	0.749



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5. Selected image segmentation results. For each image pair, the left one is by SG-GBMS and the right one is by Naive-GBMS.

SG-GBMS and Naive-GBMS is comparable, the former always converges faster than the latter does. We expect that SG-MS is applicable to general online optimization problems whose cost function is in a form of kernel sum. A study on stochastic gradient robust regression method is under investigation.

Acknowledgment

This work was supported by the following funding sources: National Natural Science Foundation Project #60518002, National Science and Technology Support Program Project #2006BAK08B06, National Hi-Tech (863) Program Project #2008AA01Z124.

References

[1] P. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In *Annual Conference on Neural Information Processing Systems*, 2007.

[2] L. Bottou. Online algorithms and stochastic approximations. In D. Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998.

[3] L. Bottou. Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004.

[4] L. Bottou and Y. L. Cun. On-line learning for very large data sets: Research articles. *Applied Stochastic Models in Business and Industry*, 21(2):137 – 151, March 2005.

[5] M. Carreira-Perpinan. Fast nonparametric clustering with gaussian blurring mean-shift. In *International Conference on Machine Learning*, 2006.

[6] M. Carreira-Perpinan. Gaussian mean-shift is an em algorithm. *IEEE Transactions on Pattern Analysis and Macnine Intelligence*, 29(5):767–776, 2007.

[7] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):790–799, 1995.

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

[9] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objets using mean shift. In *Computer Vision and Pattern Recongition,IEEE International Conference on*, volume 2, pages 142–149. IEEE, 2000.

[10] M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:471–474, Mar. 2005.

[11] N. Murata and S. ichi Amari. Statistical analysis of learning dynamics. *Signal Processing*, 74(1):3–28, March 1999.

[12] J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[13] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, Mar. 1962.

[14] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *IEEE International Conference on Computer Vision*, volume 1, pages 664–671. IEEE, 2003.

[15] X. Yuan and S. Z. Li. Half quadratic analysis for mean shift: with extension to a sequential data mode-seeking method. In *IEEE International Conference on Computer Vision*, 2007.

[16] X.-T. Yuan, B.-G. Hu, and R. He. Agglomerative mean-shift clustering via query set compression. In *SIAM International Conference on Data Mining (SDM)*, 2009.

[17] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.