Mean Shift, Mode Seeking, and Clustering

Yizong Cheng

Abstract—**Mean shift, a simple iterative procedure that shifts each data point to the average of data points in its neighborhood, is generalized and analyzed in this paper. This generalization makes some *k*-means like clustering algorithms its special cases. It is shown that mean shift is a mode-seeking process on a surface constructed with a "shadow" kernel. For Gaussian kernels, mean shift is a gradient mapping. Convergence is studied for mean shift iterations. Cluster analysis is treated as a deterministic problem of finding a fixed point of mean shift that characterizes the data. Applications in clustering and Hough transform are demonstrated. Mean shift is also considered as an evolutionary strategy that performs multistart global optimization.**

Index Terms—**Mean shift, gradient descent, global optimization, Hough transform, cluster analysis, *k*-means clustering.**

I. Introduction

Let *data* be a finite set S embedded in the n-dimensional Euclidean space, X. Let K be a *flat kernel* that is the characteristic function of the λ-ball in X,

$$K(x)=\begin{cases}1 & \text{if}\|x\|\le\lambda\\ 0 & \text{if}\|x\|>\lambda\end{cases}. \tag{1}$$

The *sample mean* at $x \in X$ is

$$m(x)=\frac{\sum_{s\in S}K(s-x)s}{\sum_{s\in S}K(s-x)}. \tag{2}$$

The difference $m(x) - x$ is called *mean shift* in Fukunaga and Hostetler [1]. The repeated movement of data points to the sample means is called the *mean shift algorithm* [1], [2]. In each iteration of the algorithm, $s \leftarrow m(s)$ is performed for all $s \in S$ simultaneously.

The mean shift algorithm has been proposed as a method for cluster analysis [1], [2], [3]. However, the intuition that mean shift is gradient ascent, the convergence of the process needs verification, and its relation with similar algorithms needs clarification.

In this paper, the mean shift algorithm is generalized in three ways. First, nonflat kernels are allowed. Second, points in data can be weighted. Third, shift can be performed on any subset of X, while the data set S stay the same.

In Section II, kernels with five operations are defined. A specific weight function unifying certain fuzzy clustering algorithms including the "maximum-entropy" clustering algorithm will be discussed. It will be shown that the k-means clustering algorithm is a limit case of mean shift.

A relation among kernels called "shadow" will be defined in Section III. It will be proved that mean shift on any kernel is equivalent to gradient ascent on the density estimated with a shadow of its. Convergence and its rate is the subject of Section IV. Section V shows some peculiar behaviors of mean shift in cluster analysis, with application in Hough transform. Section VI shows how, with a twist in weight assignment, the deterministic mean shift is transformed into a probabilistic evolutionary strategy, and how it becomes a global optimization algorithm.

II. Generalizing Mean Shift

In Section II, we first define the kernel, its notation, and operations. Then we define the generalized sample mean and the generalized mean shift algorithm. We show how this algorithm encompasses some other familiar clustering algorithms and how k-means clustering becomes a limit instance of mean shift.

A. Kernels

Definition 1. *Let* X *be the n-dimensional Euclidean space,* R^n*. Denote the* i*th component of* $x \in X$ *by* x_i*. The* norm *of* $x \in X$ *is a nonnegative number* $\|x\|$ *such that* $\|x\|^2=\sum_{i=1}^{n}|x_i|^2$*. The* inner product *of* x *and* y *in* X *is* $\langle x,y\rangle=\sum_{i=1}^{n}x_i y_i$*. A function* $K : X \to R$ *is said to be a* kernel *if there exists a* profile, $k : [0,\infty] \to R$*, such that*

$$K(x)=k\left(\|x\|^2\right) \tag{3}$$

and

1) *k is nonnegative.*
2) k is nonincreasing: $k(a) \geq k(b)$ if $a < b$.
3) k is piecewise continuous and $\int_0^\infty k(r)dr<\infty$.

Let $\alpha > 0$*. If* K *is a kernel, then* αK*,* K_α*, and* K^α *are kernels defined, respectively, as*

$$\begin{aligned}(\alpha K)(x)&=\alpha K(x),\\ K_\alpha(x)&=K\left(\frac{x}{\alpha}\right),\\ \left(K^\alpha\right)(x)&=\left(K(x)\right)^\alpha.\end{aligned} \tag{4}$$

if K *and* H *are kernels, then* $K + H$ *is a kernel defined as* $(K + H)(x) = K(x) + H(x)$ *and* KH *is a kernel defined as*

Manuscript received Aug. 27, 1993; revised Mar. 28, 1995. Recommended for acceptance by R. Duin.

Y. Cheng is with the Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, Ohio, 45221.

IEEECS Log Number P95097.

$(KH)(x) = K(x)H(x)$. These five operators can be ordered in descending precedence as K_α, K^α, αK, KH, and $K + H$. □

CLAIM 1. We have

$$\alpha(KH) = (\alpha K)H = K(\alpha H)$$
$$\alpha(K + H) = \alpha K + \alpha H$$
$$(KH)_\alpha = K_\alpha H_\alpha$$
$$(K + H)_\alpha = K_\alpha + H_\alpha$$
$$(KH)^\alpha = K^\alpha H^\alpha$$

□

EXAMPLE 1. Two kernels frequently used in this paper are the *unit flat kernel*

$$F(x) = \begin{cases} 1 & \text{if} \|x\| \le 1 \\ 0 & \text{if} \|x\| > 1 \end{cases} \tag{5}$$

and the *unit Gaussian kernel*

$$G(x) = e^{-\|x\|^2}. \tag{6}$$

These kernels are shown in Fig. 1. Clearly, the characteristic function of the λ-ball, (1), is F_λ. Also notice that $G^\beta = G_{\beta^{-1/2}}$.

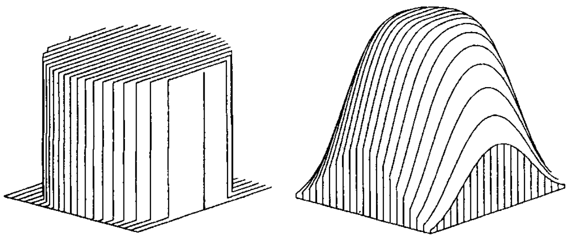


Fig. 1. (a) The flat kernel F and (b) the Gaussian kernel G.

A kernel can be "truncated" by being multiplied by a flat kernel. For example, a *truncated Gaussian kernel* is

$$\left(G^\beta F_\lambda\right)(x) = \begin{cases} e^{-\beta\|x\|^2} & \text{if} \|x\| \le \lambda \\ 0 & \text{if} \|x\| > \lambda \end{cases} \tag{7}$$

Notice that $(GF)_\lambda = G^{\lambda^{-2}} F_\lambda$. Fig. 2 shows some of the truncated Gaussian kernels. □

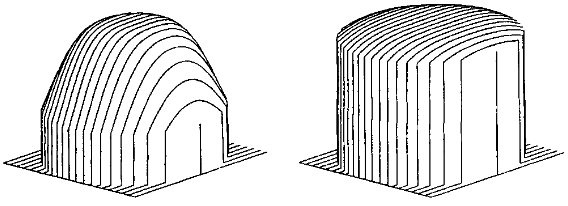


Fig. 2. Truncated Gaussian kernels (a) GF and (b) $G^{0.1}F$.

B. Mean Shift Algorithms

Now we redefine the mean shift algorithm based on our generalizations summarized in the introduction.

DEFINITION 2. *Let $S \subset X$ be a finite set (the "data" or "sample"). Let K be a kernel and $w : S \rightarrow (0, \infty)$ a* weight function. *The* sample mean *with kernel K at $x \in X$ is defined as*

$$m(x) = \frac{\sum_{s \in S} K(s - x) w(s) s}{\sum_{s \in S} K(s - x) w(s)}. \tag{8}$$

Let $T \subset X$ be a finite set (the "cluster centers"). The evolution of T in the form of iterations $T \leftarrow m(T)$ with $m(T) = \{m(t);\ t \in T\}$ is called a mean shift algorithm. *For each $t \in T$, there is a sequence t, $m(t)$, $m(m(t))$, $\cdots$, that is called the* trajectory *of t. The weight $w(s)$ can be either fixed throughout the process or re-evaluated after each iteration. It may also be a function of the current T. The algorithm halts when it reaches a fixed point ($m(T) = T$).*

When T is S, the mean shift algorithm is called a blurring process, *indicating the successive blurring of the data set, S.*□

REMARK 1. The original mean shift process proposed in [1], [3] is a blurring process, in which $T = S$. In Definition 2, it is generalized so that T and S may be separate sets with S fixed through the process, although the initial T may be a copy of S. Notice that in (8), kernel K can be replaced with kernel αK for any $\alpha > 0$, without generating any difference. This is the reason why we did not insist that $\int_X K(x)dx = 1$, which will attach a factor to K that is related to n, the dimensionality of X. Similarly, the weights $w(s)$ can be normalized so that $\sum_{s \in S} w(s) = 1$. Because of the inconsequentiality of these factors, we will use the simplest possible expressions for the kernel and the weights. We also have to assume that T is initialized such that $\sum_{s \in S} K(s - t) w(s) > 0$ for all $t \in T$. Also notice that this is a parallel algorithm, in the sense that all $t \in T$ are simultaneously updated based on the previous t and $w(s)$ values.

□

EXAMPLE 2. The "maximum entropy" clustering algorithm of Rose, Gurewitz, and Fox [4] is a mean shift algorithm when T and S are separate sets, G^β is the kernel, and

$$w(s) = \frac{1}{\sum_{t \in T} G^\beta(s - t)}, \quad s \in S. \tag{9}$$

These authors also mention that when β approaches infinity, the algorithm degenerates to k-means clustering, which is often described as an optimizing Picard iterative routine [7]:

1) Randomly initialize "cluster centers," T.

2) Compute the following function on $T \times S$:

$$v_{t,s} = \begin{cases} 1, & \text{if } t = \text{argmin}_T |s - t|^2 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

3) Update "cluster centers:"

$$t \leftarrow \frac{\sum_{s\in S} v_{t,s} s}{\sum_{s\in S} v_{t,s}} \quad t \in T. \tag{11}$$

Go to 2.

Indeed, when the profile k is strictly decreasing,

$$\frac{K^{\beta}(s-t)}{\sum_{t\in T} K^{\beta}(s-t)} \rightarrow v_{t,s}, \text{ as } \beta \rightarrow \infty. \tag{12}$$

Thus, k-means clustering is the limit of the mean shift algorithm with a strictly decreasing kernel K^{β} when $\beta \rightarrow \infty$. □

III. Mean Shift as Gradient Mapping

It has been pointed out in [1] that mean shift is a "very intuitive" estimate of the gradient of the data density. In this section, we give a more rigorous study of this intuition. Theorem 1 relates each kernel to a "shadow" kernel so that mean shift using a kernel will be in the gradient direction of the density estimate using the corresponding "shadow" kernel.

A. Shadow of a Kernel

Definition 3. *Kernel H is said to be a shadow of kernel K, if the mean shift using K,*

$$m(x) - x = \frac{\sum_{s\in S} K(s-x)w(s)s}{\sum_{s\in S} K(s-x)w(s)} - x, \tag{13}$$

is in the gradient direction at x of the density estimate using H,

$$q(x) = \sum_{s\in S} H(s-x)w(s). \tag{14}$$

□

Theorem 1. *Kernel H is a shadow of kernel K if and only if their PROFILES, h and k, satisfy the following equation.*

$$h(r) = f(r) + c\int_r^{\infty} k(t)dt, \tag{15}$$

where c > 0 is a constant and f is a piecewise constant function.

Proof. The mean shift using kernel K can be rewritten as

$$m(x) - x = \frac{1}{p(x)} \sum_{s\in S} k\left(\|s-x\|^2\right) w(s)(s-x) \tag{16}$$

with $p(x) = \sum_{s\in S} k(s-x)w(s)$, the density estimate using K.

The gradient of (14) at x is

$$\nabla q(x) = -2\sum_{s\in S} h'\left(\|s-x\|^2\right)(s-x)w(s). \tag{17}$$

To have (16) and (17) point to the same direction, we need $h'(r) = -ck(r)$ for all r and some $c > 0$. By the fundamental theorem of calculus and the requirement that $\int_0^{\infty} h(r)dr < \infty$, (15) is the only solution. In this case, we have

$$m(x) - x = \frac{\nabla q(x)}{2cp(x)}, \tag{18}$$

or, the magnitude of mean shift is in proportion to the ratio of the gradient and the local density estimate using kernel K. When a discontinuous point z is allowed in h, a constant can be added to the h from 0 to z, and $h'(r) = -ck(r)$ is still satisfied except when $r = z$. □

Claim 2. Suppose kernel H is a shadow of K, and $\alpha > 0$. The following are true.

1) αH is a shadow of K.

2) H_{α} is a shadow of K_{α}.

3) If L is a shadow of M, then $H + L$ is a shadow of $K + M$.

4) A truncated kernel KF_{α} may not be continuous at $\|x\| = \alpha$. If the shadow is also allowed to be discontinuous at the same points, then HF_{α} is a shadow of KF_{α}. □

Example 3. Using (15) we find that the *Epanechnikov kernel*

$$K(x) = \begin{cases} \left(1-\|x\|^2\right) & \text{if} \|x\| \le 1 \\ 0 & \text{if} \|x\| > 1 \end{cases} \tag{19}$$

is a shadow of the flat kernel, (5), and the *biweight kernel*

$$K(x) = \begin{cases} \left(1-\|x\|^2\right)^2 & \text{if} \|x\| \le 1 \\ 0 & \text{if} \|x\| > 1 \end{cases} \tag{20}$$

is a shadow of the Epanechnikov kernel. These kernels are so named in [2] and they are shown in Fig. 3. □

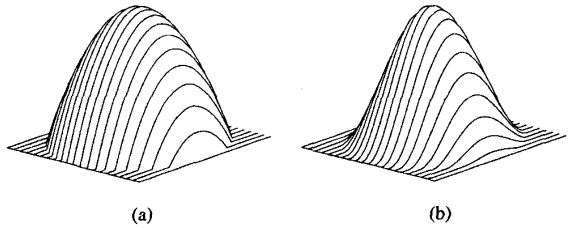


Fig. 3. (a) The Epanechnikov kernel and (b) the biweight kernel.

B. Gaussian Kernels

Theorem 2. *The only kernels that are their own shadows are the Gaussian kernel G^{β} and its truncated version $G^{\beta}F_{\lambda}$. In this case, the mean shift is equal to*

$$m(x) - x = \frac{1}{2\beta} \nabla \log q(x), \tag{21}$$

where q is the data density estimate using the same kernel.

Proof. From Theorem 1 we know that kernel K is its own shadow if and only if $k'(r) = -ck(r)$. Using the method of separation of variables, we have

$$\int \frac{dk}{k} = \int -cdr, \text{or} \log k(r) - \log k(0) = -cr .$$

This gives us $k(r) = k(0)e^{-cr}$, which makes K the Gaussian kernel. If discontinuities are allowed in k, then we have the truncated Gaussian kernel. When K is its own shadow, p in (18) is equal to q, and $(\nabla q(x))/q(x) = \nabla \log q(x)$. □

REMARK 2. A mapping $f : R^n \rightarrow R^n$ is said to be a *gradient mapping* if there exists a function $g : R^n \rightarrow R$ such that $f(x) = \nabla g(x)$ for all x [6]. Theorem 2 is a corollary of a more general result from the *symmetry principle*: f is a gradient mapping if and only if the Jacobian matrix of f is symmetric. In our case, $f(x) = m(x) - x$ and by equating $\partial f_i/\partial x_j$ and $\partial f_j/\partial x_i$ for all i and j, one obtains the necessary and sufficient condition that $k'(r) = -ck(r)$ for any mean shift to be a gradient mapping. □

C. Mode Seeking

Suppose an idealized mode in the density surface also has a Gaussian shape, which, without loss of generality, centers at the origin:

$$q(x) = e^{-\gamma\|x\|^2}. \quad (22)$$

The mean shift is now

$$m(x) - x = \frac{1}{2\beta}\nabla \log q(x) = \frac{-2\gamma x q(x)}{-2\beta q(x)} = -\frac{\gamma}{\beta}x. \quad (23)$$

Because the density surface is estimated with the kernel G^β, any mode approximated by superimposing G^β will have a $\gamma < \beta$. The mean shift (23) will not cause overshoots in this case.

Mean shift is steepest ascent with a varying step size that is the magnitude of the gradient. A notorious problem associated with steepest ascent with fixed step size is the slow movement on plateaus of the surface. For a density surface, large plateaus happen only at low density regions and after taking logarithm, the inclination of a plateau is magnified. Combined with the preceding result about overshoot avoidance, mean shift is well-adjusted steepest ascent.

IV. CONVERGENCE

Theorem 1 says that the mean shift algorithm is steepest ascent over the density of S. Each T point climbs the hill in the density surface independently. Therefore, if S or its density does not change during the execution of the algorithm, the convergence of the evolution of T is a consequence of the convergence of steepest ascent for individual T points.

However, in a blurring process, T is S, and S and its density change as the result of each iteration. In this case, convergence is not as obvious as steepest ascent. The main results of this section are two convergence theorems about the blurring process. The concepts of *radius* and *diameter* of data, defined below, will be used in the proofs.

A. Radius and Diameter of Data

DEFINITION 4. *A* direction *in X is a point on the unit sphere. That is, $a \in X$ is a direction if and only if $\|a\| = 1$. We call the mapping $\pi_a : X \rightarrow R$ with $\pi_a(x) = \langle x, a\rangle$ the* projection *in the direction of a. Let $\pi_a(S) = \{\pi_a(s); s \in S\}$. The* convex hull *of a set $Y \subset X$ is defined as*

$$h(Y) = \bigcap_{\|a\|=1} \{x \in X; \min \pi_a(Y) \le \pi_a(x) \le \max \pi_a(Y)\}. \quad (24)$$

□

CLAIM 3. The following are true.

1) $\min \pi_a(S) \le \min \pi_a(m(T)) \le \max \pi_a(m(T)) \le \max \pi_a(S)$.
2) $m(T) \subseteq h(m(T)) \subseteq h(S)$.
3) In a blurring process, we have $h(S) \supseteq h(m(S)) \supseteq h(m(m(S)))\cdots$. There exists an $x \in X$ that is in all the convex hulls of data. It is possible to make a translation so that the origin is in all the convex hulls of data. □

DEFINITION 5. *Suppose after a translation, the origin is in all the convex hulls of data during a blurring process. Then, $\rho(S) = \max \{|s|; s \in S\}$ is said to be the* radius *of data. The* diameter *of data is defined as*

$$d(S) = \sup_{\|a\|=1}\left(\max \pi_a(S) - \min \pi_a(S)\right). \quad (25)$$

□

It should be clear that $\rho(S) \le d(S) \le 2\rho(S)$. Because the convex hulls of data form a shrinking inclusion sequence, the radius or diameter of data also form a nonincreasing nonnegative sequence, which must approach a nonnegative limit. Theorem 3 says that this limit is zero when the kernel in the blurring process has a support wide enough to cover the data set.

B. Blurring With Broad Kernels

THEOREM 3. *Let k be the profile of the kernel used in a blurring process, and S_0 the initial data. If $k\left(d^2(S_0)\right) \ge \kappa$ for some $\kappa > 0$, then diameter of data approaches zero. The convergence rate is at least as fast as*

$$\frac{d(m(S))}{d(S)} \le 1 - \frac{\kappa}{4k(0)}. \quad (26)$$

PROOF. Let π be a projection, $u = \min \pi(S)$, $v = \max \pi(S)$, and $z = (u + v)/2$. Suppose

$$\sum_{s \in S, \pi(s) \le z} w(s) \ge \sum_{s \in S, \pi(s) > z} w(s). \quad (27)$$

Then, for $s \in S$,

$$\begin{aligned} v-\pi(m(s)) &= \frac{\sum_{s'\in S} K(s'-s)w(s')(v-\pi(s'))}{\sum_{s'\in S} K(s'-s)w(s')} \\ &\geq \frac{\sum_{s'\in S,\pi(s')\leq z} \kappa w(s')(v-z)}{\sum_{s'\in S} K(s'-s)w(s')} \\ &\geq \frac{\frac{1}{2}\sum_{s'\in S} w(s')\kappa \frac{v-u}{2}}{k(0)\sum_{s'\in S} w(s')} \\ &= \frac{\kappa(v-u)}{4k(0)}. \end{aligned} \tag{28}$$

Clearly, we have

$$\begin{aligned} &\max \pi(m(S)) - \min \pi(m(S)) \leq \max \pi(m(S)) - u \\ &= (v-u) - (v - \max \pi(m(S))) \leq (v-u) - \frac{\kappa(v-u)}{4k(0)} \\ &= \left(1 - \frac{\kappa}{4k(0)}\right)(v-u). \end{aligned} \tag{29}$$

The same result can be obtained when the inequality in (27) is reversed. Therefore, the result holds for all projections, and because $v - u \leq d(S)$, we completed the proof. □

EXAMPLE 4. As the simplest example, let X be the real line and $S = \{x, y\}$ with $x < y$. Let k be the profile of the kernel with $k(|x-y|^2) > 0$. If the weights on x and y are the same, a blurring process will make them converge to $(x + y)/2$ with the rate

$$m(y) - m(x) = \frac{k(0) - k(|x-y|^2)}{k(0) + k(|x-y|^2)}(y-x). \tag{30}$$

The difference between the two points will not become zero at any iteration, unless the kernel is flat in the range containing both points. When the weights are different, x and y converge to a point that may not be the weighted average of the initial x and y with the same weights.

Now assume that S is fixed through the process while T is initialized to S. Because this is no longer a blurring process, the result in Theorem 3 does not apply. That is, T may converge to more than one point. The two T points converge to $(x + y)/2$ when the density of S is unimodal. Otherwise, they will converge to the two modes of the bimodal density. When G_λ is used as the kernel, and $x = 0$, $y = 1$, the density is

$$q(z) = e^{-\frac{z^2}{\lambda^2}} + e^{-\frac{(z-1)^2}{\lambda^2}}. \tag{31}$$

The first derivative of $q(z)$ always has a zero at $z = 1/2$. But the second derivative of $q(z)$ at $z = 1/2$ is

$$\frac{2}{\lambda^2}\left(\frac{1}{\lambda^2} - 2\right)e^{-\left(\frac{1}{2\lambda}\right)^2} \tag{32}$$

which is negative when $\lambda > 1/\sqrt{2}$ and positive when $\lambda < 1/\sqrt{2}$. Thus $1/\sqrt{2}$ is a critical λ value. When λ is larger than this critical value, the density is unimodal and the two T points will converge to 1/2. When λ is smaller than $1/\sqrt{2}$, they will approach two distinct limit points. □

C. Blurring with Truncated Kernels

When truncated kernels are used in the blurring process, S may converge to many points. In fact, if the kernel is truncated so that it will not cover more than one point in S, then S will not change in the process. The result of Theorem 3 applies to an isolated group of data points that can be covered by a truncated kernel. In this case, they eventually converge to one point, although the rest of the data set may converge to other cluster centers. Again, when the kernel is not flat, no merger will take place after any finite number of iterations. (Flat kernels generate a special case where merger is possible in finite number of steps. See [8].)

When the blurring process is simulated on a digital computer, points do merge in finite number of steps, due to the limits on floating-point precision. In fact, there is a minimum distance between data points that can be maintained. Theorem 4 below shows that under this condition, the blurring process terminates in finite number of steps.

LEMMA 1. *Suppose $X = R^n$ and $r(S)$ is the radius of data in a blurring process. If the minimum distance between data points is δ, then for any direction a, there cannot be more than one data point $s \in S$ with $\pi_a(s) > r(S) - h$, where h is a function of $r(S)$, δ, and n.*

PROOF. Suppose there is a direction a such that there are $s, t \in S$ with $\pi_a(s) > r(S) - h$ and $\pi_a(t) > r(S) - h$. Let b be a direction perpendicular to a and $d_b = |\pi_b(s) - \pi_b(t)|$. Because both s and t are at least $r(S) - h$ away from the origin in direction a and one of them must also be $d_b/2$ away from the origin in direction b, the square of the radius of data cannot be smaller than $(r(S)-h)^2 + d_b^2/4$. It follows that $d_b^2 \leq 8r(S)h - 4h^2$. The distance between s and t, $\|s - t\|$, must satisfy

$$\delta^2 \leq \|s-t\|^2 \leq (n-1)(8r(S)h - 4h^2) + h^2.$$

If

$$(n-1)(8r(S)h - 4h^2) + h^2 > \delta^2,$$

then these s and t cannot exist. This condition is satisfied when

$$h < h_m = \left[4(n-1)r(S) - \sqrt{16(n-1)^2 r^2(S) - (4n-5)\delta^2}\right] \Big/ (4n-5)$$

□

LEMMA 2. *In a blurring process, suppose the minimum distance between data points is δ, $\kappa > 0$ is a constant such that when $K(x) > 0$, $K(x) > \kappa k(0)$, and*

$$W = \frac{\min_{s \in S} w(s)}{\sum_{s \in S} w(s)} \tag{33}.$$

The radius of data, $r(S)$, reaches its final value in no more than $r(S)/(\kappa W h_m)$ steps, where h_m is defined in Lemma 1.

PROOF. Let $s \in S$ be a data point that moves during an iteration. Then there must have been another data point $s' \in S$ such that $K(s - s') > \kappa k(0)$. Let a be a direction. Lemma 1 says that at least one of these two points, s or s', denoted with s'', must have $\pi_a(s'') \leq r(S) - h_m$. Hence,

$$\begin{aligned} r(S) - \pi_a(m(s)) &= \frac{\sum_{t \in S} K(t-s)w(t)(r(S) - \pi_a(t))}{\sum_{t \in S} K(t-s)w(t)} \\ &\geq \frac{K(s''-s)w(s'')(r(S) - \pi_a(s''))}{\sum_{t \in S} k(0)w(t)} \\ &\geq \kappa Wh. \end{aligned} \tag{34}$$

This shows that all moving points in S moves in one iteration to a position at least κWh_m away from the current $r(S)$. Therefore, if $r(S)$ changes during an iteration, its change must be at least κWh_m. □

THEOREM 4. *If data points cannot move arbitrarily close to each other, and $K(x)$ is either zero or larger than a fixed positive constant, then the blurring process reaches a fixed point in finitely many iterations.*

PROOF. Lemma 2 says that the radius of data reaches its final value in finite number of steps. Lemma 2 also implies that those points at this final radius will not affect other data points or each other. Hence, they can be taken out from consideration for further process of the algorithm. The same argument can be applied to the rest of data. Since S is finite, by induction on the size of S, a fixed point must be reached in finitely many iterations. More precisely, the blurring process halts in no more than $r(S)/(\kappa Wh_m)$ steps. □

REMARK 3. It is important that when close data points are merged, the radius of data does not increase. A simple mechanism is merging close points into one of them. This must be done before the exhaustion of floating-point precision, because truncation may indeed increase the radius of data and cause cycles. An extreme case is the blurring process applied to categorical data, when a flat kernel based on the Hamming distance and round-off to the nearest integer are used in each mean shift step. In this case, the mean shift step becomes

$$s \leftarrow \text{Majority}\{t \in S; \|t - s\| \leq \lambda\}. \tag{35}$$

Round-off may actually shift a data point out of the flat kernel centering at it. For example, 0111, 1011, 1101, and 1110 all have Hamming distance 3 from 0000, but the majority of the five will be 1111, which has a distance 4 from the center, 0000, and there is a projection on which max $\pi(S)$ actually increases. □

D. Fixed Points as Local Maxima

In Section III, we showed that mean shift for individual points in X is hill climbing on the data density function. Because the data density function also evolves in the blurring process, it is difficult to see where the hill climbing leads to. When the evolving set of points in X, either S (in blurring) or T (as cluster centers), is treated as a point in X^N, where N is the number of points involved, real functions can be constructed and the fixed points of mean shift can be identified as local maxima of these functions.

THEOREM 5. *When S is fixed, the stable fixed points of the mean shift process*

$$T \leftarrow m(T) = \left\{ m(t) = \frac{\sum_{s \in S} K(s-t)w(s)s}{\sum_{s \in S} K(s-t)w(s)} ; t \in T \right\} \tag{36}$$

are local maxima of

$$U(T) = \sum_{s \in S} w(s) \sum_{t \in T} H(t-s), \tag{37}$$

where H is a shadow of K. For the blurring process, $S \leftarrow m(S)$, assuming weights of data points do not change, the fixed points are the local maxima of

$$V(S) = \sum_{s,t \in S} H(s-t)w(s)w(t). \tag{38}$$

PROOF. When S is fixed, each $t \in T$ reaches its fixed point when $\nabla q(t) = 0$, using the result and notation in (18). Because $U(T) = \sum_{t \in T} q(t)$, a local maximum of U is reached when each $t \in T$ attains a local maximum of $q(t)$. Because local minima of q are not stable fixed points for t, a stable fixed point of U can only be its local maximum. For the blurring process, we have

$$\frac{\partial V}{\partial s} = 2\sum_{t \in S} h'\left(\|s-t\|^2\right)(s-t)w(s)w(t). \tag{39}$$

Notice that $w(s)$ will not change and thus is treated as a constant. $\partial V / \partial s = 0$ is equivalent to

$$\sum_{t \in S} K(s-t)w(t)(t-s) = 0 \tag{40}$$

or

$$m(s) - s = \frac{\sum_{t \in S} K(s-t)w(t)(t-s)}{\sum_{t \in S} K(s-t)w(t)} = 0, \tag{41}$$

and thus, the local maxima of $V(S)$ are fixed points of $S \leftarrow m(S)$. By the same reason as before, they are the only stable fixed points. □

V. Mean Shift Clustering

Theorem 3 shows that in a blurring process with a broad kernel (one with a broad support), data points do converge to a single position. To the other extreme, when the kernel is truncated to the degree that it covers no more than one data point at any position, the initial data set is a fixed point of the blurring process and thus no merger takes place. When the kernel size is between these extremes, data points may have trajectories that merge into varying numbers of "cluster centers."

Iteration of mean shift gives rise to natural clustering algorithms. The final T contains the final positions of the cluster centers. In k-means like clustering, T is not initialized to S, fuzzy membership or nearest center classification must be used to decide how data points are divided into clusters.

In this section, we will study the clustering results when T is initialized to S or when T is S (the blurring process). The data set S is partitioned into clusters based solely on their mean shift trajectories. When two data points or their T representatives converge to the same final position, they are considered to belong to the same cluster. Unlike k-means like clustering algorithms, which are probabilistic because the randomness of the initialization of T, mean shift clustering with T initialized to S is deterministic.

A. Clustering as a Natural Process

Many clustering algorithms are treated as means for optimizing certain measures about the partitioning. For example, the k-means clustering algorithm is aiming at minimizing the within-group sum of squared errors [7], and the maximum entropy clustering algorithm is to maximize entropy while the within-group sum of squared errors is held constant. Sometimes, it is the algorithm itself that is emphasized, for instance in the case of the k-means clustering. The initial cluster centers, T, are randomly or strategically chosen, and there is no guarantee that any execution of the algorithm will reach the global minimum. After the execution of the algorithm, all one can say is that a local minimum is reached, and the optimization goal becomes illusive.

At other times, the reach of a global optimum is essential. The maximum entropy clustering is an example. The actual iteration that hopefully attains the goal is de-emphasized, based on precisely the same reason, that every run of the algorithm only reaches a local maximum [5]. It is known that optimization problems like these are NP-hard [9]. Hence, in general, clustering as optimization is computationally unattainable.

The philosophy of this paper is that clustering can also be viewed as the result of some natural process, like mean shift or blurring. As a deterministic process, the clustering result provides a characteristic of the data set. In the light of Theorem 5, when T is initialized to S, the final configuration of T is a local maximum of $U(T)$; when T is S, it is a local maximum of $V(S)$. The result or purpose of mean shift clustering is to use a local maximum of U or V as a characterization of S. The global maximum of U or V is not only unattainable, but also undesirable. For instance, V reaches its global maximum when S shrinks to one point, which is the result only when the kernel has a broad support. The number and distribution of local maxima of U and V depend only on the kernel, the dimensionality of the space, and the data size.

Example 5. To visualize mean shift as clustering, we randomly chose an S of size 100 in the unit square (Fig. 4a) and applied five different variations of mean shift. Processes were terminated when no update larger than 10^{-4} took place. The truncated Gaussian kernel, $(GF)_{\beta^{-1/2}}$, was used in Figs. 4b, 4c, and 4d, while the Gaussian kernel, "nontruncated," G^{β}, was used in Figs. 4e) and 4f, all with $\beta = 30$. The blurring process was used in Fig. 4b and nonblurring, meaning

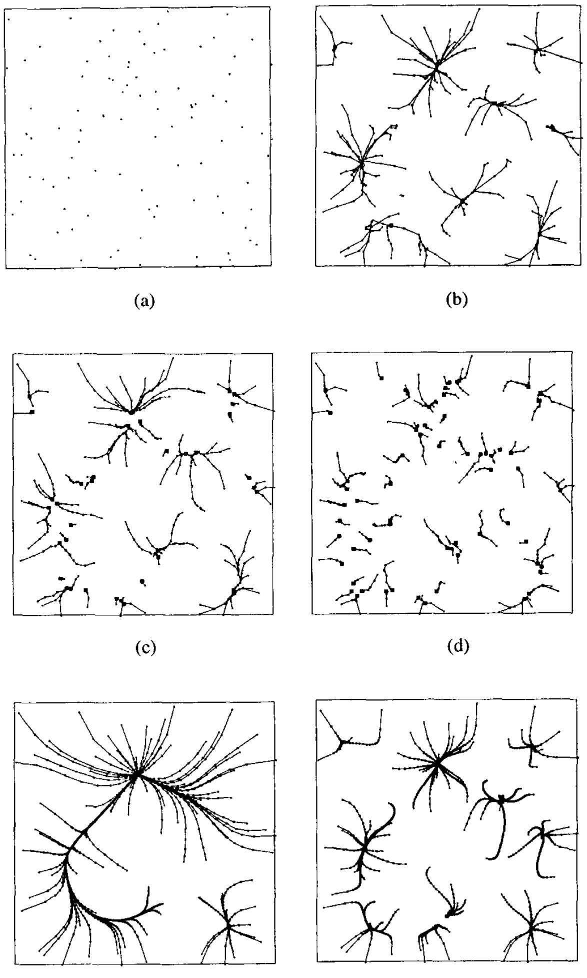


Fig. 4. Trajectories of different mean shift variations on the same data set. (a) The data set S, also the initial T set for nonblurring processes. (b) A blurring process with 10 iterations and nine clusters. (c) A nonblurring mean shift process with a truncated Gaussian kernel and uniform weights. It terminated in 20 iterations at 37 clusters. (d) Nonblurring mean shift with a truncated Gaussian kernel and an adaptive weight function. It terminated in 20 iterations at 64 clusters. (e) Nonblurring with a nontruncated Gaussian kernel and uniform weights. It terminated in 274 iterations at two clusters. (f) Nonblurring with a nontruncated Gaussian kernel and adaptive weights. It terminated in 127 iterations with 13 clusters.

T was only initialized to S but S is fixed, was used in others. In (b), (c), and (e), data points were equally weighted, while in (d) and (f), a weight w that satisfies $1/w(s) = \sum_{t \in T} K(t-s)$ was used. □

B. Validation of Clustering

When the truncated Gaussian kernel $(GF)_{\beta^{-1/2}}$ with varying β is used, we can see that in general, the smaller β is, the fewer clusters will be generated. But this may not always be true. Furthermore, a smaller cluster associated with a larger β value may not be the result of a "split" of a larger cluster associated with a smaller β value. However, if the clustering outcomes are plotted against a range of β values, then it will be clear that some clustering outcomes are transient and unstable while others are more stable. The recurrence of a clustering outcome with varying β values can be used as an indication that the pattern may be more valid than the transient ones.

EXAMPLE 6. Fig. 5 is a plot of the blurring process applied on the velocities of 82 galaxies [11]. The same data were fit with a bimodal normal mixture model by Roeder [10] and conclusion was that the observable universe contained two superclusters of galaxies surrounded by large voids. However, by observing Fig. 5, one can see that the most stable outcome is three instead of two clusters. This shows that while bimodal mixture or k-means clustering requires some prior guessing about the number of clusters, the result from mean shift clustering is less arbitrary. □

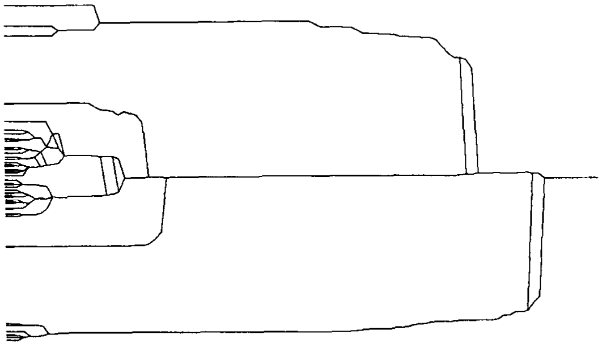


Fig. 5. Clustering of 82 galaxies based on their velocities. The tree-shape diagram shows the relative validity of clustering outcomes with the kernel $(GF)_\lambda$ of different λ values. Larger λ values were used near the root of the tree, while smaller λ values were used near the leaves of the tree, where the dimension across the tree indicates the velocities.

EXAMPLE 7. Another example of blurring within a parameter continuum is demonstrated in Fig. 6, where 62 alphanumeric characters were treated as data points in a 144 (the number of pixels involved in the 8 × 18 font) dimensional Euclidean space. Blurring with kernel $(GF)_\lambda$ with λ ranging from 1.6 to 3.8 was performed. The clustering results were discretized and displayed as two-tone pixels. □

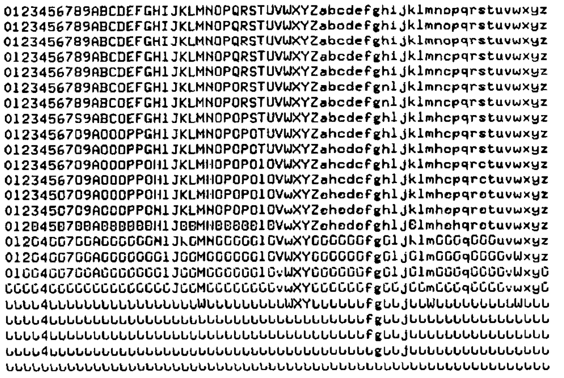
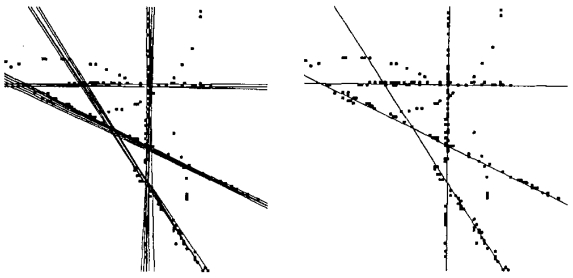


Fig. 6. Blurring of 62 8 × 18 font alphanumeric characters. Each row is the outcome of blurring using the kernel $(GF)_\lambda$ with a λ value between 1.6 and 3.8. The average number of iterations was 4.7.

C. Application to Hough Transform

EXAMPLE 8. Fig. 7 shows an application of mean shift clustering in generalized Hough transform. 300 edge pixels were randomly chosen from a 100 × 100 image and each pair of them generated a straight line passing them. The intersections of these lines and the borders of the image are rounded off to the nearest integers (pixel positions) and they are registered with 400 × 400 accumulators in the parameter space, similar to the "muff" transform suggested by Wallace [12]. □



(a) (b)

Fig. 7. Mean shift peak detection in the parameter space of Hough (muff) transform. The image contains 300 edge pixels. Pairs of edge pixels make 44,850 suggestions of possible lines that are coded as points in the parameter space. Each line is coded with two integers between 0 and 400, that label the intersections of the line with the borders of the image. (a) shows the line detection based on the number of suggestions falling into the each point in the parameter space. A threshold of 30 was used. (b) is the result after mean shift is applied to these lines in the parameter space. The kernel was $(GF)_{10}$, weights were the number of suggestions, and four clusters emerged as the four lines shown.

VI. MEAN SHIFT OPTIMIZATION

The blurring process moves data points in the gradient direction of the function q on X,

$$q(x) = \sum_{s \in S} K(s-x)w(s). \tag{42}$$

In clustering with mean shift, this function q is considered as an approximation to the data density, and mean shift finds the local maxima of the density function.

When data S is uniformly distributed in an area of X, and $w(s)$ is the value of a *target function* $f : X \rightarrow R$ at the point s, q becomes an approximation of f with some scaling. Mean shift with this setting will find *all* the local maxima of f in a region, and this leads to another application of mean shift, namely global optimization.

Because now the initial data set S has to be randomly (and hopefully uniformly) generated, the algorithm becomes probabilistic, even when T is S or is initialized to S. To compensate the inevitable non-uniformity of any finite random set, the weight $w(s)$ can be the f value at s augmented with a data density balancing factor, as

$$w(w) = \frac{f(s)}{\sum_{t \in S} K(t-s)}. \tag{43}$$

When the blurring process is used, the next generation of S will concentrate more at the modes of the approximated F function, but the weight w will contain a factor that offsets this effect. It is also possible to make the algorithm more deterministic by placing the initial S on regular grid points.

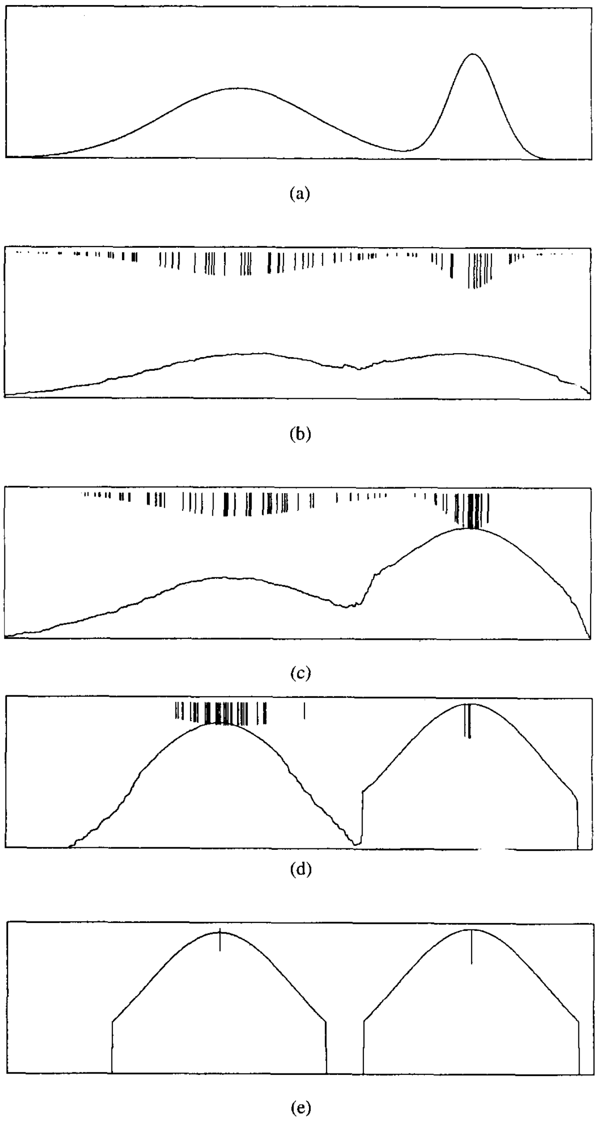


Fig. 8. Multistart global optimization using blurring. (a) shows the function f, whose global maximum is to be found. The next four figures show the mean shift of S, at the (b) initial, (c) first, (d) third, and (e) fifth iterations of a blurring process when f is used as the weight function. In each of these four figures, the vertical bars show the positions and f values of the S points, and the curve shows the q function, whose local maxima locations approximate those of f.

EXAMPLE 9: Fig. 8 is a demonstration of this optimization strategy. (a) is the underlying function f, whose maxima are to be found. This function is unknown, but with a price, $f(x)$ may be obtained for any $x \in X$. The upper half of (b) shows the initial randomly chosen 100 $x \in X$, as the set S, along with their $f(x)$ values. The lower half is the estimated f function using

$$q(x) = \sum_{s \in S} (GF)_\lambda (s-x) w(s),$$

$$w(s) = \frac{f(s)}{\sum_{t \in S} (GF)_\lambda (t-s)}. \tag{44}$$

In this demonstration, the range of X is the unit interval and $\lambda = 0.18$. (c) contains the S set along with $f(s)$ values after a single mean shift step, with the estimated f using this data set S in the lower half. (d) and (e) are the snapshots after three and five mean shift steps respectively. After five iterations, the global maximum and a local maximum were discovered and S reached its final configuration, a fixed point of the blurring process. □

Mean shift optimization is a parallel hill climbing method comparable to many genetic algorithms. The blurring process in effect is an evolutionary strategy, with f being the so-called fitness function.

Mean shift optimization also shows some similarity with some multistart global optimization methods [13]. Because it is necessary to have sample points on both sides of a local maximum in order for mean shift to work, the method may not be efficient in a space of high dimensionality or an f function with too many local maxima. Nevertheless, it has the unique property that the same simple iteration works for both local and global optimization, compared to most multistart methods where separate local and global approaches are used.

VII. CONCLUDING REMARKS

Suppose one has made a number of observations, performed a series of experiments, or collected a stack of cases. What is a natural way in which the memory of data is organized? It could be organized as a sorted list based on some key, or a decision tree, or a rule-based system, or a distributed associative memory. But what can be more basic than to associate each experience with similar ones, and to extract the common features that make them different from others? Mean shift, or confusing a point with the average of similar points, must be a simple and natural process that plays a role in memory organization. A multiresolution representation of the data, containing both commonality and specificity, seems likely to be the foundation of law discovery and knowledge acquisition.

Since this process is so intuitive and basic, it should have been used as an ingredient in various modeling and algorithmic efforts, or at least it should have been studied in mathematics as a simple dynamic system. However, based on the author's search and communication, the existence of previous efforts is not apparent. To the best knowledge of the author, Fukunaga and Hostetler [1] is still the first work proposing mean shift explicitly as an iterative algorithm, and a rigorous and comprehensive treatment of the process has not been done.

This paper attempted to provide an appropriate generalization to the mean shift algorithm, so that many interesting and useful properties would be preserved. One property is that the process is either a gradient mapping, or a similar one that seeks modes of a real function. Compared to gradient descent or ascent methods, mean shift seems more effective in terms of adapting to the right step size.

The two major applications of mean shift discussed in this paper, namely cluster analysis and global optimization, have not been practiced widely. There may be computational obstacles, and they may not be suitable for problems with prohibitive sizes and dimensionalities. The computational cost of an iteration of mean shift is $O(n^2)$ where n is the size of S, the data set. It is obviously possible to reduce this time complexity to $O(n \log n)$, by a better storage of the data, when only neighboring points are used in the computation of the mean.

ACKNOWLEDGMENTS

The author thanks Keinosuke Fukunaga and John Schlipf for conversations and Zhangyong Wan for early exploration of the convergence of the process.

This work was supported by the National Science Foundation, under Grant no. IRI-9010555.

REFERENCES

[1] K. Fukunaga and L.D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Information Theory*, vol. 21, pp. 32-40, 1975.

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

[3] Y. Cheng and K.S. Fu, "Conceptual clustering in knowledge organization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, pp. 592-598, 1985.

[4] K. Rose, E. Gurewitz, and G.C. Fox, "Statistical mechanics and phase transitions in clustering," *Physical Review Letters*, vol. 65, pp. 945-948, 1990.

[5] K. Rose, E. Gurewitz, and G.C. Fox, "Constrained clustering as an optimization method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.15, pp. 785-794, 1993.

[6] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. San Diego: Academic Press, 1970.

[7] S.Z. Selim and M.A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 81-86, 1984.

[8] Y. Cheng and Z. Wan, "Analysis of the blurring process," *Computational Learning Theory and Natural Learning Systems*, vol. 3, T. Petsche, et al., eds., 1992.

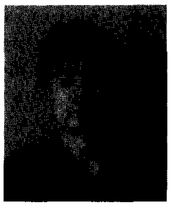
[9] P. Brucker, "On the complexity of clustering problems," Henn, Korte, and Oettli, eds. *Optimization and Operations Research*. Berlin: Springer-Verlag, 1978.

[10] K. Roeder, "Density estimation with confidence sets exemplified by superclusters and voids in the galaxies," *J. Amer. Statistical Assoc.*, vol. 85, pp. 617-624, 1990, also see [11].

[11] B.S. Everitt, *Cluster Analysis*. 3rd ed., London: Edward Arnold, 1993.

[12] R.S. Wallace, "A modified Hough transform for lines," *IEEE CVPR Conf.*, pp. 665-667, San Francisco, 1985.

[13] A.H.G. Rinnooy Kan and G.T. Timmer, "Stochastic global optimization methods Part I: Clustering methods," *Mathematical Programming*, vol. 39, pp. 27-56, 1987.



Yizong Cheng received the BSEE and PhD degrees from Purdue University, West Lafayette, Ind., in 1981 and 1986, respectively. He is now on the faculty of the University of Cincinnati.

His current research interests are in data self-organization and knowledge discovery.