Gaussian mean shift is an EM algorithm

Miguel Á. Carreira-Perpiñán
Dept. of Computer Science & Electrical Engineering
OGI School of Science & Engineering, Oregon Health & Science University
20000 NW Walker Road, Beaverton, OR 97006, USA
Email: miguel@csee.ogi.edu

August 23, 2006

Abstract

The mean-shift algorithm, based on ideas proposed by Fukunaga and Hostetler (1975), is a hill-climbing algorithm on the density defined by a finite mixture or a kernel density estimate. Mean-shift can be used as a nonparametric clustering method and has attracted recent attention in computer vision applications such as image segmentation or tracking. We show that, when the kernel is Gaussian, mean-shift is an expectation-maximisation (EM) algorithm, and when the kernel is non-gaussian, mean-shift is a generalised EM algorithm. This implies that mean-shift converges from almost any starting point and that, in general, its convergence is of linear order. For Gaussian mean-shift we show: (1) the rate of linear convergence approaches 0 (superlinear convergence) for very narrow or very wide kernels, but is often close to 1 (thus extremely slow) for intermediate widths, and exactly 1 (sublinear convergence) for widths at which modes merge; (2) the iterates approach the mode along the local principal component of the data points from the inside of the convex hull of the data points; (3) the convergence domains are nonconvex and can be disconnected and show fractal behaviour. We suggest ways of accelerating mean-shift based on the EM interpretation.

Keywords: mean-shift algorithm, Gaussian mixtures, kernel density estimators, EM algorithm, clustering

1 Mode-finding and the mean shift algorithm

The modes of a density $p(\mathbf{x})$ are important from a machine learning point of view. As an example for clustering [10, 11, 16], the density may represent the distribution of data in a given problem and then the modes may be taken as representatives of clusters. As an example for multivalued regression [4], the modes of the conditional distribution $p(\mathbf{y}|\mathbf{z})$, where $\mathbf{x} = (\mathbf{y}, \mathbf{z})$, may be used to represent a multivalued mapping $\mathbf{z} \rightarrow \mathbf{y}$. It is then necessary to design algorithms that will find modes of a density. In principle, any maximisation algorithm may be applied, using multiple starting points to find as many different modes as possible. However, for a widespread class of densities, that of finite mixtures—which includes kernel density estimates and Gaussian mixtures—one can derive a particularly simple fixed-point algorithm, mean-shift (described below). The mean-shift algorithm has proven particularly successful in image segmentation [11]. Here, one first builds a kernel density estimate for the image pixels and declares each of its modes as representative of a cluster. Then, a given pixel is assigned to the mode it converges to under the mean-shift algorithm. The method is deterministic (since mean-shift has no step size) and nonparametric (because of the kernel density estimate); it poses no a priori assumptions about the number or shape of the clusters. Our main objective in this paper is to show that mean-shift is a particular instance of a well-known class of algorithms, that of (generalised) expectation-maximisation (EM); and to study aspects of its convergence rate and convergence domains.

The mean-shift algorithm can be derived as follows. Given M D-dimensional data points $\{\boldsymbol{\mu}_m\}_{m=1}^M \subset \mathbb{R}^D$, consider a kernel density estimate [28] with kernel $K(t)$ for $t \geq 0$:

$$p(\mathbf{x}) = \sum_{m=1}^{M} p(m)p(\mathbf{x}|m) = \sum_{m=1}^{M} \pi_m \frac{1}{Z_m} K\left(d(\mathbf{x}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma}_m)\right) \tag{1}$$

where $p(m) = \pi_m \in (0, 1)$ is the weight or mixing proportion of point m (satisfying $\sum_{m=1}^M \pi_m = 1$), $\boldsymbol{\Sigma}_m$ is its covariance matrix (positive definite), Z_m is a normalisation constant that only depends on $\boldsymbol{\Sigma}_m$ (e.g. $Z_m =$

$|2\pi\boldsymbol{\Sigma}_m|^{1/2}$ for the Gaussian kernel), and $d(\mathbf{x}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma}_m) = (\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m)$ is the Mahalanobis distance. For example, we have the kernels

$$\text{Gaussian: } K(t) = e^{-t/2} \qquad \text{Epanechnikov: } K(t) = \begin{cases} 1-t, & t \in [0,1) \\ 0, & t \geq 1 \end{cases} \qquad \text{Student's } t\text{: } K(t) = \left(1 + \frac{t}{\alpha}\right)^{-\frac{\alpha+D}{2}}.$$

To find a mode of $p(\mathbf{x})$ we can seek stationary points

$$\frac{\partial p(\mathbf{x})}{\partial \mathbf{x}} = 2 \sum_{m=1}^{M} \pi_m \frac{1}{Z_m} K'\left(d(\mathbf{x}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma}_m)\right) \boldsymbol{\Sigma}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m) = \mathbf{0}$$

(where $K' = dK/dt$) and solving for $\mathbf{x}$ suggests the fixed-point iteration scheme $\mathbf{x}^{(\tau+1)} = \mathbf{f}(\mathbf{x}^{(\tau)})$ where

$$\mathbf{f}(\mathbf{x}) = \left(\sum_{m=1}^{M} \pi_m \frac{1}{Z_m} K'\left(d(\mathbf{x}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma}_m)\right) \boldsymbol{\Sigma}_m^{-1} \right)^{-1} \sum_{m=1}^{M} \pi_m \frac{1}{Z_m} K'\left(d(\mathbf{x}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma}_m)\right) \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\mu}_m. \tag{2}$$

For a homoscedastic estimator ($\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}\ \forall m$) this simplifies to

$$\mathbf{f}(\mathbf{x}) = \sum_{m=1}^{M} \frac{\pi_m K'\left(d(\mathbf{x}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma})\right)}{\sum_{m'=1}^{M} \pi_{m'} K'\left(d(\mathbf{x}, \boldsymbol{\mu}_{m'}; \boldsymbol{\Sigma})\right)} \boldsymbol{\mu}_m. \tag{3}$$

In practice the most common use is for a homoscedastic estimator with constant weights ($\pi_m = \frac{1}{M}\ \forall m$) and isotropic covariances ($\boldsymbol{\Sigma}_m = \sigma^2 \mathbf{I}$), which further simplifies to

$$\mathbf{f}(\mathbf{x}) = \sum_{m=1}^{M} \frac{K'\left(\left\| \frac{\mathbf{x} - \boldsymbol{\mu}_m}{\sigma} \right\|^2 \right)}{\sum_{m'=1}^{M} K'\left(\left\| \frac{\mathbf{x} - \boldsymbol{\mu}_{m'}}{\sigma} \right\|^2 \right)} \boldsymbol{\mu}_m. \tag{4}$$

This fixed-point iteration scheme is called the *mean-shift algorithm*, where the vector $\mathbf{f}(\mathbf{x}) - \mathbf{x}$ is the mean shift (which is proportional to the gradient of $p(\mathbf{x})$ for isotropic kernels). Unlike other optimisation algorithms, mean-shift does not use a step size parameter, and effectively defines a mapping $\mathbf{f}^\infty = \mathbf{f} \circ \mathbf{f} \circ \dots$ that maps points in $\mathbb{R}^D$ to stationary points. By applying the mean-shift algorithm to each of the data points $\boldsymbol{\mu}_m$ (or indeed any point $\mathbf{x} \in \mathbb{R}^D$) we can determine its mode of convergence and then cluster together points converging to the same mode. The number of clusters is thus determined automatically for given σ.

When the kernel is Gaussian, $K' \propto K$ and we get a particularly simple and elegant algorithm (where, by Bayes' theorem, $p(m|\mathbf{x}) = p(\mathbf{x}|m)p(m)/p(\mathbf{x})$ is the posterior probability or responsibility of component m given point $\mathbf{x}$):

$$\mathbf{f}(\mathbf{x}) = \left(\sum_{m=1}^{M} p(m|\mathbf{x}) \boldsymbol{\Sigma}_m^{-1} \right)^{-1} \sum_{m=1}^{M} p(m|\mathbf{x}) \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\mu}_m. \tag{5}$$

For isotropic kernels this simplifies to

$$\mathbf{f}(\mathbf{x}) = \sum_{m=1}^{M} q(m|\mathbf{x}) \boldsymbol{\mu}_m \qquad q(m|\mathbf{x}) = \frac{p(m|\mathbf{x}) \sigma_m^{-2}}{\sum_{m'=1}^{M} p(m'|\mathbf{x}) \sigma_{m'}^{-2}} \tag{6}$$

where the $q(m|\mathbf{x})$ values are the responsibilities $p(m|\mathbf{x})$ reweighted by the inverse variance and renormalised. This further simplifies for homoscedastic isotropic kernels to

$$\mathbf{f}(\mathbf{x}) = \sum_{m=1}^{M} p(m|\mathbf{x}) \boldsymbol{\mu}_m \qquad p(m|\mathbf{x}) = \frac{\pi_m e^{-\frac{1}{2} \left\| \frac{\mathbf{x} - \boldsymbol{\mu}_m}{\sigma} \right\|^2}}{\sum_{m'=1}^{M} \pi_{m'} e^{-\frac{1}{2} \left\| \frac{\mathbf{x} - \boldsymbol{\mu}_{m'}}{\sigma} \right\|^2}} \tag{7}$$

where the new point $\mathbf{x}^{(\tau+1)} = \mathbf{f}(\mathbf{x}^{(\tau)})$ is the conditional mean of the mixture under the current point $\mathbf{x}^{(\tau)}$. Note that, for non-gaussian kernels, $p(m|\mathbf{x})$ involves K while $\mathbf{f}(\mathbf{x})$ (and thus the iteration) involves K'; see eqs. (3)–(4).

The mean-shift algorithm is so simple that it has probably been discovered many times. In 1975, Fukunaga and Hostetler [16] were perhaps the first to propose its idea and also introduced the term "mean shift". Since 1981, the algorithm was also independently known in the statistics literature as *mean update algorithm* (see [29]

pp. 167ff and references therein). Fukunaga and Hostetler derived the algorithm for the Epanechnikov kernel (for computational convenience, since it has finite support) as gradient ascent on $\log p(\mathbf{x})$ with a variable step size, without proving convergence. Their version is different from the one we give here: they suggested that every data point $\boldsymbol{\mu}_m$ should move according to its mean shift $\mathbf{f}(\boldsymbol{\mu}_m) - \boldsymbol{\mu}_m$ at each iteration till convergence. This version was called "blurring process" by Cheng [10], who discussed the convergence of the blurring process and suggested the use of the mean-shift algorithm without moving the data points (as considered here). Carreira-Perpiñán [3], motivated by the problem of finding all the modes of a Gaussian mixture, independently rediscovered the algorithm for the Gaussian kernel and proved its convergence for arbitrary covariance matrices [5]. Comaniciu and Meer [11] gave a different convergence proof for certain isotropic kernels, including the Gaussian and Epanechnikov kernel (the latter converging in a finite number of iterations owing to its finite support). They applied mean-shift to image filtering and image segmentation, unleashing a wave of interest in the algorithm for these and other computer vision applications, such as tracking (e.g. [12, 13, 30]). Fashing and Tomasi [15] related the mean-shift algorithm for isotropic homoscedastic kernels to bound optimisation and showed it coincides with Newton's method when K' is piecewise constant. Algorithms similar to mean-shift have appeared in scale-space clustering [9, 25, 32, 33], in clustering by deterministic annealing [26] and in pre-image finding in kernel-based methods [27].

The contributions of this paper are as follows. First, we show that Gaussian mean-shift (eq. (5)) is an expectation-maximisation (EM) algorithm (section 2) and that non-gaussian mean-shift (eq. (2)) is a generalised EM algorithm (section 3). This means that in both cases convergence is assured from almost every starting point. Previous convergence proofs [11, 15] were not based on the connection with EM and were restricted to the isotropic homoscedastic case ($\boldsymbol{\Sigma}_m = \sigma^2\mathbf{I}\ \forall m$); our proof holds for the general case where each component can have its own, full covariance. The EM view also draws attention to two separate steps in the Gaussian mean-shift iteration: the computation of posterior probabilities (E step) and the average of the data with respect to them (M step). Next, we focus on Gaussian mean-shift with isotropic kernels. In section 4 we characterise the speed of convergence as a function of the kernel width and show that: (1) the convergence is typically linear, as expected with EM algorithms, and so can be very slow; (2) the convergence becomes superlinear for very narrow or very wide kernels, and sublinear when modes merge; (3) the iterates approach the mode they converge to along the local principal component of the data points, from within the convex hull of the data points. In section 5 we show that the convergence domains (thus the clusters) are generally nonconvex and can be disconnected and, remarkably, fractal-like. We conclude by suggesting ways of accelerating the mean-shift algorithm based on its view as an EM algorithm.

2 Gaussian mean shift as an EM algorithm

Here we show that the Gaussian mean-shift algorithm can be derived as an *expectation-maximisation (EM) algorithm* [14, 18]. The idea of this proof was suggested to the author by Chris Williams (University of Edinburgh) and appeared originally in references [5, 8]. The idea consists of defining an artificial maximum likelihood problem where (1) the model is a Gaussian mixture given by the original kernel density estimate (thus the kernel centroids are given by the data set to which Gaussian mean-shift is applied), but this mixture can be moved around rigidly (thus the model has a single parameter, the displacement vector $\mathbf{v}$), and (2) the sample to which this model is fit contains a single point at $\mathbf{x} = \mathbf{0}$. Then, we can prove that any maximum likelihood estimate of the displacement vector is a mode of the original kernel density estimate; and the corresponding EM algorithm coincides with Gaussian mean-shift.

Assume $\{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^M$ are the Gaussian-mixture parameters for a Gaussian mean-shift problem as in eqs. (1) and (5). Consider the following density model for $\mathbf{x} \in \mathbb{R}^D$ with parameter $\mathbf{v} = (v_1, \dots, v_D)^T$ and where $\{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^M$ are fixed:

$$q(\mathbf{x}|\mathbf{v}) = \sum_{m=1}^{M} \pi_m \left|2\pi\boldsymbol{\Sigma}_m\right|^{-\frac{1}{2}} e^{-\frac{1}{2}d(\mathbf{x},\boldsymbol{\mu}_m-\mathbf{v};\boldsymbol{\Sigma}_m)}. \tag{8}$$

That is, $\mathbf{x}|\mathbf{v}$ is a D-dimensional Gaussian mixture where component m has mixing proportion π_m (fixed), mean vector $\boldsymbol{\mu}_m - \mathbf{v}$ ($\boldsymbol{\mu}_m$ fixed) and covariance matrix $\boldsymbol{\Sigma}_m$ (fixed). Varying $\mathbf{v}$ results in a rigid translation of the whole mixture as a block rather than the individual components varying separately. Now consider fitting this model by maximum likelihood to a data set $\{\mathbf{x}_n\}_{n=1}^N$ and let us derive an EM algorithm to estimate the parameters $\mathbf{v}$. The data set $\{\mathbf{x}_n\}_{n=1}^N$ is arbitrary at this stage, and later we particularise it to a single point at the origin; note this data set is different from the collection of centroids $\{\boldsymbol{\mu}_m\}_{m=1}^M$ from the mean-shift problem. Call $z_n \in \{1, \dots, M\}$ the (unknown) index of the mixture component that generated data point $\mathbf{x}_n$. Then:

E step The complete-data log-likelihood, as if all $\{z_n\}_{n=1}^N$ were known, and assuming iid data, is

$\sum_{n=1}^{N} \mathcal{L}_{n,\text{complete}}(\mathbf{v}) = \sum_{n=1}^{N} \log q(\mathbf{x}_n, z_n|\mathbf{v})$ and so its expectation with respect to the current posterior distribution is

$$\begin{aligned} Q(\mathbf{v}|\mathbf{v}^{(\tau)}) &= \sum_{n=1}^{N} \mathrm{E}_{q(z_n|\mathbf{x}_n,\mathbf{v}^{(\tau)})} \{\mathcal{L}_{n,\text{complete}}(\mathbf{v})\} \\ &= \sum_{n=1}^{N} \sum_{z_n=1}^{M} q(z_n|\mathbf{x}_n, \mathbf{v}^{(\tau)}) \log\{q(z_n|\mathbf{v}) q(\mathbf{x}_n|z_n, \mathbf{v})\} \\ &= \sum_{n=1}^{N} \sum_{z_n=1}^{M} q(z_n|\mathbf{x}_n, \mathbf{v}^{(\tau)}) \log q(\mathbf{x}_n|z_n, \mathbf{v}) + C \end{aligned} \tag{9}$$

where the term $C = \sum_{n=1}^{N} \sum_{z_n=1}^{M} q(z_n|\mathbf{x}_n, \mathbf{v}^{(\tau)}) \log \pi_{z_n}$ is independent of $\mathbf{v}$.

M step The new parameter estimates $\mathbf{v}^{(\tau+1)}$ are obtained from the old ones $\mathbf{v}^{(\tau)}$ as $\mathbf{v}^{(\tau+1)} = \arg\max_{\mathbf{v}} Q(\mathbf{v}|\mathbf{v}^{(\tau)})$. To perform this maximisation, we equate the gradient of Q with respect to $\mathbf{v}$ to zero:

$$\frac{\partial Q}{\partial \mathbf{v}} = \sum_{n=1}^{N} \sum_{z_n=1}^{M} q(z_n|\mathbf{x}_n, \mathbf{v}^{(\tau)}) \frac{1}{q(\mathbf{x}_n|z_n, \mathbf{v})} \frac{\partial q(\mathbf{x}_n|z_n, \mathbf{v})}{\partial \mathbf{v}} = \mathbf{0}. \tag{10}$$

As a function of $\mathbf{v}$, $q(\mathbf{x}_n|z_n, \mathbf{v})$ is a Gaussian density of mean $\boldsymbol{\mu}_{z_n} - \mathbf{x}_n$ and covariance $\boldsymbol{\Sigma}_{z_n}$, so we get

$$\frac{\partial q(\mathbf{x}_n|z_n, \mathbf{v})}{\partial \mathbf{v}} = q(\mathbf{x}_n|z_n, \mathbf{v}) \boldsymbol{\Sigma}_{z_n}^{-1} (\boldsymbol{\mu}_{z_n} - \mathbf{x}_n - \mathbf{v})$$

and solving for $\mathbf{v}$ in eq. (10) results in

$$\mathbf{v}^{(\tau+1)} = \left(\sum_{n=1}^{N} \sum_{z_n=1}^{M} q(z_n|\mathbf{x}_n, \mathbf{v}^{(\tau)}) \boldsymbol{\Sigma}_{z_n}^{-1} \right)^{-1} \sum_{n=1}^{N} \sum_{z_n=1}^{M} q(z_n|\mathbf{x}_n, \mathbf{v}^{(\tau)}) \boldsymbol{\Sigma}_{z_n}^{-1} (\boldsymbol{\mu}_{z_n} - \mathbf{x}_n).$$

If now we choose the data set as simply containing the origin (i.e., $N = 1$ and $\mathbf{x}_1 = \mathbf{0}$), rename z_1 as m, and note that $q(m|\mathbf{0}, \mathbf{v}) = p(m|\mathbf{v})$, we obtain the M step as:

$$\mathbf{v}^{(\tau+1)} = \left(\sum_{m=1}^{M} p(m|\mathbf{v}^{(\tau)}) \boldsymbol{\Sigma}_m^{-1} \right)^{-1} \sum_{m=1}^{M} p(m|\mathbf{v}^{(\tau)}) \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\mu}_m \tag{11}$$

which is formally identical to the iterative scheme of eq. (5). The log-likelihood $\mathcal{L}(\mathbf{v}) = \sum_{n=1}^{N} \log q(\mathbf{x}_n|\mathbf{v})$ is

$$\log q(\mathbf{0}|\mathbf{v}) = \log \sum_{m=1}^{M} \pi_m \left|2\pi\boldsymbol{\Sigma}_m\right|^{-\frac{1}{2}} e^{-\frac{1}{2} d(\mathbf{v}, \boldsymbol{\mu}_m; \boldsymbol{\Sigma}_m)} = \log p(\mathbf{v}).$$

In summary, the Gaussian mean-shift iterative scheme (5) is equivalent to the EM algorithm on the Gaussian-mixture model (8) when trying to fit a sample at the origin.

General properties of the EM algorithm for Gaussian mixtures tell us that convergence is assured for almost any starting point and is of linear order [14, 18, 24, 34]. Specifically, at every iteration τ, the iterative scheme (11) will either increase the log-likelihood or leave it unchanged, and correspondingly, the iterative scheme (5) will monotonically increase the density value $p(\mathbf{x})$ or leave it unchanged. We can show directly that $p(\mathbf{v}^{(\tau+1)}) \geq p(\mathbf{v}^{(\tau)})$ as follows (this derivation mirrors that of the general EM algorithm; again, we use $p(\cdot)$ rather than $q(\cdot)$ for clarity, noting that $q(\mathbf{0}|\mathbf{v}) = p(\mathbf{v})$, etc.):

$$\begin{aligned} \log p(\mathbf{v}^{(\tau+1)}) - \log p(\mathbf{v}^{(\tau)}) &= \log \sum_{m=1}^{M} \frac{\pi_m p(\mathbf{v}^{(\tau+1)}|m)}{p(\mathbf{v}^{(\tau)})} \overset{\text{(a)}}{=} \log \sum_{m=1}^{M} \frac{\pi_m p(\mathbf{v}^{(\tau+1)}|m)}{p(\mathbf{v}^{(\tau)}|m)\pi_m} p(m|\mathbf{v}^{(\tau)}) \\ &\overset{\text{(b)}}{\geq} \sum_{m=1}^{M} p(m|\mathbf{v}^{(\tau)}) \log \frac{\pi_m p(\mathbf{v}^{(\tau+1)}|m)}{\pi_m p(\mathbf{v}^{(\tau)}|m)} = Q(\mathbf{v}^{(\tau+1)}|\mathbf{v}^{(\tau)}) - Q(\mathbf{v}^{(\tau)}|\mathbf{v}^{(\tau)}) \overset{\text{(c)}}{\geq} 0 \end{aligned} \tag{12}$$

where we have applied (a) Bayes' theorem to $p(\mathbf{v}^{(\tau)})$, (b) Jensen's inequality ($a_m,\ b_m > 0$: $\log \sum a_m b_m \geq \sum a_m \log b_m$) and (c) the fact that $\mathbf{v}^{(\tau+1)}$ maximises $Q(\mathbf{v}|\mathbf{v}^{(\tau)})$. Besides, also by Jensen's inequality in (b), the increase in log-likelihood is strictly positive except when $\mathbf{v}^{(\tau+1)} = \mathbf{v}^{(\tau)}$ (i.e., at a fixed point). Thus, $\mathbf{v}^{(\tau+1)} \neq \mathbf{v}^{(\tau)} \Rightarrow p(\mathbf{v}^{(\tau+1)}) > p(\mathbf{v}^{(\tau)})$.

In principle, convergence can occur to a saddle point or to a minimum instead of to a mode (in the very unlikely case where the initial value is at a minimum, the scheme will remain stuck at it). Since both saddle points and minima are unstable for maximisation, a small random perturbation will cause the EM algorithm to diverge from them. Thus, practical convergence will almost always be to a mode.

There is an interesting relation with the result of Fashing and Tomasi [15], who considered mean-shift in the homoscedastic isotropic case ($\boldsymbol{\Sigma}_m = \sigma^2\mathbf{I}\ \forall m$) and showed that the mean-shift iteration maximises a quadratic lower bound $\rho(\mathbf{v})$ on the density $p(\mathbf{v})$. For the Gaussian kernel, their quadratic lower bound is given by eq. (15) in [15], which is (in our notation)

$$\rho(\mathbf{v}) = a - \frac{p(\mathbf{v}^{(\tau)})}{2\sigma^2} \sum_{m=1}^{M} p(m|\mathbf{v}^{(\tau)}) \left\|\mathbf{v} - \boldsymbol{\mu}_m\right\|^2$$

where $a > 0$ is constant (independent of $\mathbf{v}$), and the bound verifies $\rho(\mathbf{v}^{(\tau)}) = p(\mathbf{v}^{(\tau)})$ and $\rho(\mathbf{v}) \leq p(\mathbf{v})\ \forall \mathbf{v} \in \mathbb{R}^D$. The EM algorithm uses a quadratic lower bound very similar to that one but applied to $\log p(\mathbf{v})$ instead (i.e., $p(\mathbf{v})$ is lower bounded by a Gaussian). Defining $\varrho(\mathbf{v})$ as follows:

$$\varrho(\mathbf{v}) = \log p(\mathbf{v}^{(\tau)}) - Q(\mathbf{v}^{(\tau)}|\mathbf{v}^{(\tau)}) + Q(\mathbf{v}|\mathbf{v}^{(\tau)})$$
$$Q(\mathbf{v}|\mathbf{v}^{(\tau)}) = a_1 - \frac{1}{2\sigma^2} \sum_{m=1}^{M} p(m|\mathbf{v}^{(\tau)}) \left\|\mathbf{v} - \boldsymbol{\mu}_m\right\|^2$$

where $a_1 > 0$ is constant, from eq. (12) we have that $\varrho(\mathbf{v}^{(\tau)}) = \log p(\mathbf{v}^{(\tau)})$ and $\varrho(\mathbf{v}) \leq \log p(\mathbf{v})\ \forall \mathbf{v} \in \mathbb{R}^D$. While the bounds $\rho(\mathbf{v})$ and $\varrho(\mathbf{v})$ are different, both are quadratic and are maximised at $\mathbf{v}^{(\tau)}$. For non-gaussian kernels, the bound of [15] is still quadratic while the Q function is not quadratic anymore (see next section).

It is also possible to view the Gaussian mean-shift algorithm (for the homoscedastic isotropic case) as a dynamical system, where τ is a continuous variable and the path of $\mathbf{v}(\tau)$ is a continuous curve, and prove that it converges (appendix A).

3 Non-gaussian mean shift as a generalised EM algorithm

For an arbitrary kernel $K(t)$, $t \geq 0$ we define

$$q(\mathbf{x}|\mathbf{v}) = \sum_{m=1}^{M} \pi_m \frac{1}{Z_m} K\left(d(\mathbf{x}, \boldsymbol{\mu}_m - \mathbf{v}; \boldsymbol{\Sigma}_m)\right) \tag{13}$$

analogously to eq. (8). The derivation of the EM algorithm proceeds as for the Gaussian case up to eq. (10), but now

$$\frac{\partial q(\mathbf{x}_n|z_n, \mathbf{v})}{\partial \mathbf{v}} = \frac{2}{Z_{z_n}} K'\left(d(\mathbf{x}, \boldsymbol{\mu}_{z_n} - \mathbf{v}; \boldsymbol{\Sigma}_{z_n})\right) \boldsymbol{\Sigma}_{z_n}^{-1}(\boldsymbol{\mu}_{z_n} - \mathbf{x}_n - \mathbf{v}).$$

Considering as before $N = 1$ and $\mathbf{x}_1 = \mathbf{0}$, renaming variables, and using $p(\cdot)$ instead of $q(\cdot)$ we obtain

$$\frac{\partial Q}{\partial \mathbf{v}} = \frac{2}{p(\mathbf{v})} \sum_{m=1}^{M} \frac{p(m|\mathbf{v}^{(\tau)})}{p(m|\mathbf{v})} \frac{\pi_m}{Z_m} K'\left(d(\boldsymbol{\mu}_m, \mathbf{v}; \boldsymbol{\Sigma}_m)\right) \boldsymbol{\Sigma}_m^{-1}(\mathbf{v} - \boldsymbol{\mu}_m) = \mathbf{0}$$

where we have used the expression (from Bayes' theorem)

$$p(m|\mathbf{v}) = \frac{\pi_m}{p(\mathbf{v})} \frac{1}{Z_m} K\left(d(\boldsymbol{\mu}_m, \mathbf{v}; \boldsymbol{\Sigma}_m)\right).$$

For the Gaussian kernel, $K' \propto K$ and we recover eq. (11) by solving exactly for $\mathbf{v}$, but for non-gaussian kernels we cannot solve exactly for $\mathbf{v}$. We have two options:

Fixed I			
ϵ	I	outer	inner
10^{-1}	100	304	342
10^{-2}	100	257	531
10^{-3}	100	246	801
10^{-4}	100	239	1107
10^{-5}	100	232	1445
10^{-6}	100	226	1812
10^{-7}	100	220	2198
10^{-8}	100	213	2618

Fixed ϵ			
ϵ	I	outer	inner
$\mathbf{10^{-8}}$	**1**	**314**	**314**
10^{-8}	2	237	473
10^{-8}	3	220	650
10^{-8}	4	215	834
10^{-8}	5	214	1019
10^{-8}	10	213	1852
10^{-8}	15	213	2501
10^{-8}	100	213	2618

Table 1: Total number of iterations spent in the outer and inner loops of the generalised EM algorithm of section 3, as a function of the tolerance ϵ for convergence in $\mathbf{u}$ (i.e., $\left\|\mathbf{u}^{(\nu+1)}-\mathbf{u}^{(\nu)}\right\| < \epsilon$) and the maximum number of iterations I allowed in the inner loop. The problem consisted of $M = 594$ points in $D = 3$ dimensions with a Student's t kernel ($\alpha = 1$). In all cases, the initial point $\mathbf{v}^{(0)}$ was the same and the outer loop was run till $\left\|\mathbf{v}^{(\tau+1)}-\mathbf{v}^{(\tau)}\right\| < 10^{-8}$. The computational cost is proportional to the number of inner-loop iterations, so the fastest version is for $I = 1$, corresponding to the mean-shift algorithm, which took 314 iterations (boldface). Similar results were obtained from other initial points and in other problems.

- To do an exact M step, we can use the following fixed-point iteration scheme *for fixed* $\mathbf{v}^{(\tau)}$:

$$\mathbf{u}^{(\nu+1)} = \mathbf{g}(\mathbf{u}^{(\nu)}) \quad \text{with} \qquad \mathbf{g}(\mathbf{u}) = \left(\sum_{m=1}^{M} \rho_m(\mathbf{u})\boldsymbol{\Sigma}_m^{-1}\right)^{-1} \sum_{m=1}^{M} \rho_m(\mathbf{u})\boldsymbol{\Sigma}_m^{-1}\boldsymbol{\mu}_m \tag{14}$$

  where

$$\rho_m(\mathbf{u}) = \frac{p(m|\mathbf{v}^{(\tau)})}{p(m|\mathbf{u})}\frac{\pi_m}{Z_m}K'\left(d(\boldsymbol{\mu}_m,\mathbf{u};\boldsymbol{\Sigma}_m)\right).$$

  Thus, the final EM algorithm consists of two nested loops: the outer one is the usual iteration over τ; the inner one is the iteration over ν for fixed τ. The inner loop is run till $\mathbf{u}^{(\nu)}$ has approximately converged.

- To do an inexact M step where the inner loop is run only a few times, stopping before $\mathbf{u}^{(\nu)}$ converges. This is a *generalised EM (GEM) algorithm*. In a GEM, in the M step we choose a $\mathbf{u}$ that simply increases Q rather than maximising it. As is evident from eq. (12), convergence is guaranteed as for the EM algorithm [14, 18] as long as the iteration (14) on ν increases Q.

  If in eq. (14) we choose as initial $\mathbf{u}$ (for $\nu = 0$) $\mathbf{u}^{(0)} = \mathbf{v}^{(\tau)}$ and do a single step then we obtain that $\mathbf{v}^{(\tau+1)} = \mathbf{u}^{(1)}$ coincides with the mean-shift iteration of eq. (2). Thus, *the original mean-shift algorithm for non-gaussian kernels is a generalised EM algorithm*.

We can define a version of the GEM algorithm where the inner loop is initialised at $\mathbf{u}^{(0)} = \mathbf{v}^{(\tau)}$ and run till either the inner loop converges within a tolerance ϵ (i.e., $\left\|\mathbf{u}^{(\nu+1)}-\mathbf{u}^{(\nu)}\right\| < \epsilon$) or the inner loop runs for I iterations. This establishes a continuum between both extreme cases: for $\epsilon \to 0$ and $I \to \infty$ we obtain the exact M step (thus the EM algorithm), while for $I = 1$ we obtain the mean-shift algorithm. We have experimented with this and have found that over a range of ϵ and I values, mean-shift is faster overall. That is, while one can reduce the number of iterations of the outer loop by doing a few inner-loop iterations, the total number of inner-loop iterations remains higher. Since the inner-loop iterations are a bit more costly (because of the computation of K as well as K' and the additional products), it seems that the mean-shift algorithm is always faster. Table 1 shows a representative example.

In summary, we have shown that the mean-shift algorithm is a generalised EM algorithm for non-gaussian kernels. Thus, its convergence is guaranteed (as long as the iteration (14) on ν increases Q) and its order of convergence is linear in general [18], though other orders of convergence are possible in special cases (one being the Epanechnikov kernel, for which convergence occurs in a finite number of iterations).

4 Speed of convergence of Gaussian mean shift

In section 2 we saw that Gaussian mean-shift is an EM algorithm, and thus that it converges to a mode from almost any starting point, monotonically increasing the density value $p(\mathbf{x})$ or leaving it unchanged. The order of

convergence of EM algorithms is generally linear [18, p. 105]. Here we study in detail the speed of convergence and the character of the iterates' path for mean-shift in the homoscedastic, isotropic case ($\boldsymbol{\Sigma}_m = \sigma^2\mathbf{I}\ \forall m$), i.e., we focus on the iterative scheme (7). We ignore the degenerate case where all points are coincident ($\boldsymbol{\mu}_m = \boldsymbol{\mu}_1\ \forall m$), for which the density is Gaussian and convergence is achieved in a single step.

Recall [22, pp. 28ff] that if a sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$ converges to $\mathbf{x}^*$ then the convergence is said to be of first order, or linear, if there exists a positive constant $r < 1$ (the rate of convergence) such that, for all τ sufficiently large, $\left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^*\right\| / \left\|\mathbf{x}^{(\tau)} - \mathbf{x}^*\right\| \leq r$. The convergence is superlinear if $\lim_{\tau\to\infty} \left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^*\right\| / \left\|\mathbf{x}^{(\tau)} - \mathbf{x}^*\right\| = 0$ and sublinear if $\lim_{\tau\to\infty} \left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^*\right\| / \left\|\mathbf{x}^{(\tau)} - \mathbf{x}^*\right\| = 1$.

Let $\mathbf{x}^*$ be a mode of $p(\mathbf{x})$ to which the mean-shift sequence $\{\mathbf{x}^{(\tau)}\}_{\tau=0}^{\infty}$ converges. Taylor-expanding $\mathbf{f}$ around $\mathbf{x}^*$ to first order and using $\mathbf{f}(\mathbf{x}^*) = \mathbf{x}^*$ we obtain:

$$\mathbf{x}^{(\tau+1)} = \mathbf{f}(\mathbf{x}^{(\tau)}) = \mathbf{x}^* + \mathbf{J}(\mathbf{x}^*)(\mathbf{x}^{(\tau)} - \mathbf{x}^*) + \mathcal{O}(\|\mathbf{x}^{(\tau)} - \mathbf{x}^*\|^2) \tag{15}$$

and thus

$$\mathbf{x}^{(\tau+1)} - \mathbf{x}^* \approx \mathbf{J}(\mathbf{x}^*)(\mathbf{x}^{(\tau)} - \mathbf{x}^*). \tag{16}$$

$\mathbf{J}(\mathbf{x}^*)$ is the $D \times D$ Jacobian of $\mathbf{f}$ evaluated at $\mathbf{x}^*$. The Jacobian can be computed to be

$$\mathbf{J}(\mathbf{x}) = \frac{1}{\sigma^2}\left(\sum_{m=1}^{M} p(m|\mathbf{x})\boldsymbol{\mu}_m\boldsymbol{\mu}_m^T - \mathbf{f}(\mathbf{x})\mathbf{f}(\mathbf{x})^T\right) = \frac{1}{\sigma^2}\sum_{m=1}^{M} p(m|\mathbf{x})(\boldsymbol{\mu}_m - \mathbf{f}(\mathbf{x}))(\boldsymbol{\mu}_m - \mathbf{f}(\mathbf{x}))^T$$

which is symmetric positive (semi)definite. Since at least some of its diagonal elements are positive (unless all points $\boldsymbol{\mu}_m$ are coincident) we have $\mathbf{J}(\mathbf{x}) \neq \mathbf{0}\ \forall \mathbf{x} \in \mathbb{R}^D$. The Hessian of $p(\mathbf{x})$ is given by

$$\mathbf{H}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sigma^2}(\mathbf{J}(\mathbf{x}) - \mathbf{I}). \tag{17}$$

This result can be obtained by direct manipulation, or by using the ratio of information matrices (see appendix B). Therefore, $\mathbf{H}$ and $\mathbf{J}$ have the same eigenvectors and their respective eigenvalues μ_d and λ_d are related by

$$\mu_d = \frac{p(\mathbf{x})}{\sigma^2}(\lambda_d - 1),\ d = 1, \ldots, D. \tag{18}$$

At a mode $\mathbf{x}^*$ we have $\mathbf{x}^* = \mathbf{f}(\mathbf{x}^*)$ and $\mathbf{H}(\mathbf{x}^*)$ is negative definite or negative semidefinite. Consider first the generic case where $\mathbf{H}(\mathbf{x}^*)$ is negative definite. Since $\mathbf{J}(\mathbf{x}^*)$ is positive definite, from eq. (18) we obtain that $\lambda_d \in (0, 1)$, confirming that the sequence $\{\mathbf{x}^{(\tau)}\}_{\tau=0}^{\infty}$ indeed converges to $\mathbf{x}^*$. (In fact, $\mathbf{x}^*$ and $\sigma^2\mathbf{J}(\mathbf{x}^*)$ are the mean and covariance of the points $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M$ weighted by $p(m|\mathbf{x}^*)$.) Since $\mathbf{J}(\mathbf{x}^*) \neq \mathbf{0}$, the convergence is linear and has a rate $r \in (0, 1)$ typically[1] equal to the largest eigenvalue of $\mathbf{J}(\mathbf{x}^*)$:

$$r = \lim_{\tau\to\infty} \frac{\left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^*\right\|}{\left\|\mathbf{x}^{(\tau)} - \mathbf{x}^*\right\|} = \lambda_{\max} < 1.$$

We can consider the following cases for σ:

- $\sigma \to 0$: this is the case of widely separated components, where the mode $\mathbf{x}^*$ almost coincides with the closest component centroid $\boldsymbol{\mu}_{m^*}$. We have $p(m|\mathbf{x}^*) \to \delta_{mm^*}$, $\mathbf{x}^* \to \boldsymbol{\mu}_{m^*}$, $\mathbf{J}(\mathbf{x}^*) \to \mathbf{0}$ and the convergence becomes superlinear. However, the convergence is never quadratic because $\mathbf{J}(\mathbf{x}^*) \neq \mathbf{0}\ \forall \sigma > 0$.
- $\sigma \to \infty$: this is the case of very wide components, where there is a unique mode $\mathbf{x}^*$ almost coinciding with the mean of the data $\overline{\mathbf{x}} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\mu}_m$. We have $p(m|\mathbf{x}^*) \to \frac{1}{M}$, $\mathbf{x}^* \to \overline{\mathbf{x}}$, $\mathbf{J}(\mathbf{x}^*) \approx \frac{1}{\sigma^2}\operatorname{cov}\{\boldsymbol{\mu}_m\} \to \mathbf{0}$ and $p(\mathbf{x}) \to \mathcal{N}(\overline{\mathbf{x}}, \sigma^2\mathbf{I})$. The convergence becomes superlinear again but never quadratic.
- Intermediate σ: this is the practically useful case where the density estimate is most accurate, but convergence is slowest. The convergence is linear with rate $r = \lambda_{\max}$. To get an idea of the value of r, consider the setup in fig. 1 where we have several points arranged in a uniform grid (in image segmentation using spatial and range features, this is representative of a uniform region in the image, where intensity or colour are constant). The figure shows the convergence rate r as a function of σ for the mode at the grid centre. For a range of σ, the density is almost flat and r is almost 1, which means an extremely slow convergence rate (Newton's method, however, converges accurately in a few iterations). The slow convergence in practice of Gaussian mean-shift has been noted by Carreira-Perpiñán [3] and Comaniciu and Meer [11].

[1] The rate r need not always equal the largest eigenvalue of $\mathbf{J}(\mathbf{x}^*)$ [19]. For Gaussian mixtures with symmetry with respect to a minor eigenvector of $\mathbf{J}(\mathbf{x}^*)$ associated with eigenvalue $\lambda_i < \lambda_{\max}$, initial iterates lying exactly along that eigenvector will converge at a faster rate λ_i along that minor coomponent. However, in practice convergence will typically be at the slowest rate $\lambda_{\max}$, along the principal component.

Consider now the non-generic case where $\mathbf{H}(\mathbf{x}^*)$ is negative semidefinite. This happens when modes merge[2] or disappear at a given value of σ, since the Hessian changes sign there. For example, for a mixture with two identical components $p(x) = \frac{1}{2}\mathcal{N}(x; -1, \sigma) + \frac{1}{2}\mathcal{N}(x; 1, \sigma)$, the origin $x = 0$ is a mode for $\sigma \geq 1$. For $\sigma > 1$ the Hessian is negative definite: $H(0) < 0$. But for $\sigma = 1$ the Hessian is negative semidefinite: $H(0) = 0$ and $J(0) = 1$ (the mapping in this case is $f(x) = \tanh x$). Thus when the Hessian is negative semidefinite the asymptotic rate is $r = 1$ and the convergence is *sublinear*. (The convergence is still guaranteed: theorem 3.2 in p. 88 of [18] states that an EM sequence converges to a stationary point of the likelihood if the $Q(\mathbf{v}|\mathbf{v}^{(\tau)})$ function is continuous in $\mathbf{v}$ and $\mathbf{v}^{(\tau)}$, which it obviously is in our case.) From eq. (15), the update $\mathbf{x}^{(\tau+1)} - \mathbf{x}^{(\tau)}$ along those directions associated with $\lambda_{\max} = 1$ is $\mathcal{O}(\|\mathbf{x}^{(\tau)} - \mathbf{x}^*\|^2)$ or of even a higher order, and thus exceedingly slow[3]. In practice this results in Gaussian mean-shift taking a much larger number of iterations for those widths σ where clusters merge (or almost merge). This situation does occur with image segmentation; a dramatic demonstration appears in fig. 4 of [6].

The path followed by the mean-shift iterates has the following properties (illustrated in fig. 2):

- Near convergence, the path follows the direction of the eigenvector $\mathbf{u}_{\max}$ associated with $\lambda_{\max}$, because eq. (16) applied k times yields

$$\mathbf{x}^{(\tau+k)} - \mathbf{x}^* \approx \mathbf{J}(\mathbf{x}^*)^k(\mathbf{x}^{(\tau)} - \mathbf{x}^*) \approx \lambda_{\max}^k(\mathbf{u}_{\max}^T(\mathbf{x}^{(\tau)} - \mathbf{x}^*))\,\mathbf{u}_{\max}$$

  and so $\mathbf{x}^{(\tau)} - \mathbf{x}^*$ is in the direction of $\mathbf{u}_{\max}$. Since $\sigma^2\mathbf{J}(\mathbf{x}^*)$ is the local covariance at $\mathbf{x}^*$, then the mean-shift iterates follow its principal component.

- As shown by Carreira-Perpiñán [3] for the Gaussian case, and evident from eqs. (2)–(3), in both the homoscedastic and the isotropic cases and for any type of kernel (not necessarily Gaussian), each iterate is a convex linear combination of the data points (as are all the stationary points, including the modes), and so the path lies in the interior of the convex hull of the data points. (In general for finite mixtures of densities from the exponential family, the EM algorithm always stays in the convex hull of a certain set of parameters [24, eq. (5.3)].) However, this is not true for non-isotropic kernels (including diagonal kernels), where both the iterates and the modes may lie outside the convex hull of the data points (for an example, see fig. 1 in [8]).

- As shown by Comaniciu and Meer [11], the path is smooth in the sense that consecutive steps (consecutive mean-shift vectors $\mathbf{f}(\mathbf{x}) - \mathbf{x}$) always make an angle in $(-\frac{\pi}{2}, \frac{\pi}{2})$.

- The step sizes are not the best along the mean-shift direction. While a line search [22, chapter 3] would produce a better $\mathbf{x}^{(\tau+1)}$, it is doubtful this would reduce the overall computational cost (even if the line search is inexact) because of the need to evaluate $p(\mathbf{x})$ several times—each evaluation is $\mathcal{O}(M)$ and the number of data points M is usually large. Besides, the convergence rate would remain linear. We suggest other ways to accelerate Gaussian mean-shift in section 6.

In summary, Gaussian mean-shift converges nearly always linearly, approaches superlinear convergence when $\sigma \to 0$ or $\sigma \to \infty$, and converges sublinearly at mode merges. The practically useful cases require an intermediate σ for which the rate r of linear convergence can be close to 1, thus convergence will be very slow. The mean-shift iterates smoothly approach the mode along the principal component of the weighted covariance matrix of the data points, from within the convex hull of the data points.

5 Convergence domains of Gaussian mean shift

The number of modes of the kernel density estimate determines the number of clusters. Thus, one would expect that M points should produce at most M modes (since additional modes are an artifact of the kernel rather than a sign of meaningful structure in the data). However, this is not necessarily so. Consider first the 1D case. It is possible to prove that the Gaussian kernel is the only kernel for which the kernel density estimate has at most M

[2]Note that, from Morse theory [20], even in this case the modes of a Gaussian mixture, and indeed all its stationary points, are always isolated; thus we can never have a ridge of modes.

[3]It is interesting to compare this with another situation where EM has been shown to become very slow as well, that of linear models such as independent component analysis in the low-noise case [2, 23]. Here, the EM updates to the model parameters are proportional to the noise variance and thus very small. This result is counterintuitive, since with low noise the model is a better fit to the data and so one might have expected the convergence to be faster (as we have seen, for Gaussian mean-shift the convergence does speed up when $\sigma \to 0$). This emphasises the need for a careful study of the convergence rate for each particular EM algorithm.

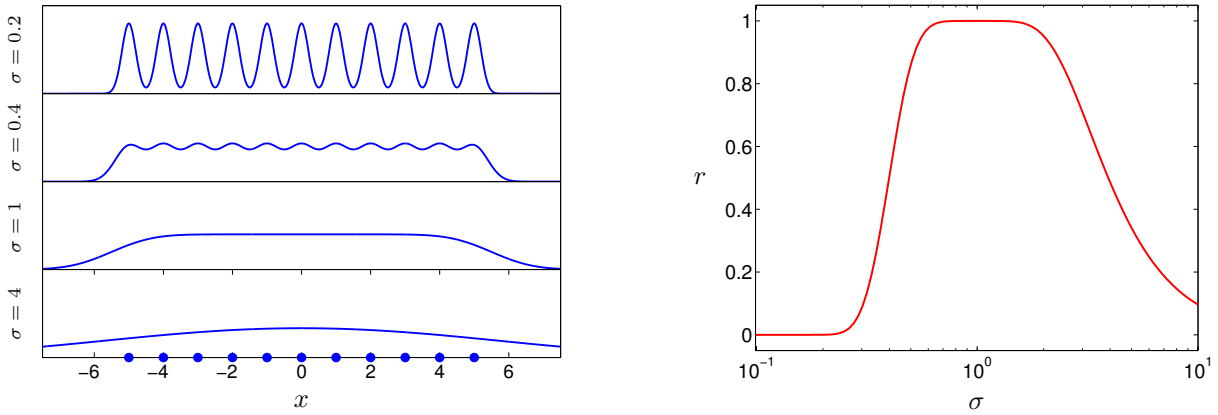

Figure 1: Empirical determination of the rate of linear convergence for the mean-shift algorithm with Gaussian, isotropic, homoscedastic kernels. $M = 11$ points are located on a 1D grid symmetric around $x = 0$. The left panel shows the kernel density estimate $p(x)$ for 4 different kernel widths σ. The right panel shows the convergence rate $r = J(0) \in (0, 1)$ for the mode at the origin as a function of σ. While $r \to 0$ for small or large σ (superlinear convergence), $r \approx 1$ for a wide range of intermediate σ, corresponding to locally flat densities. In such cases, which are frequent in practice, convergence can be very slow.

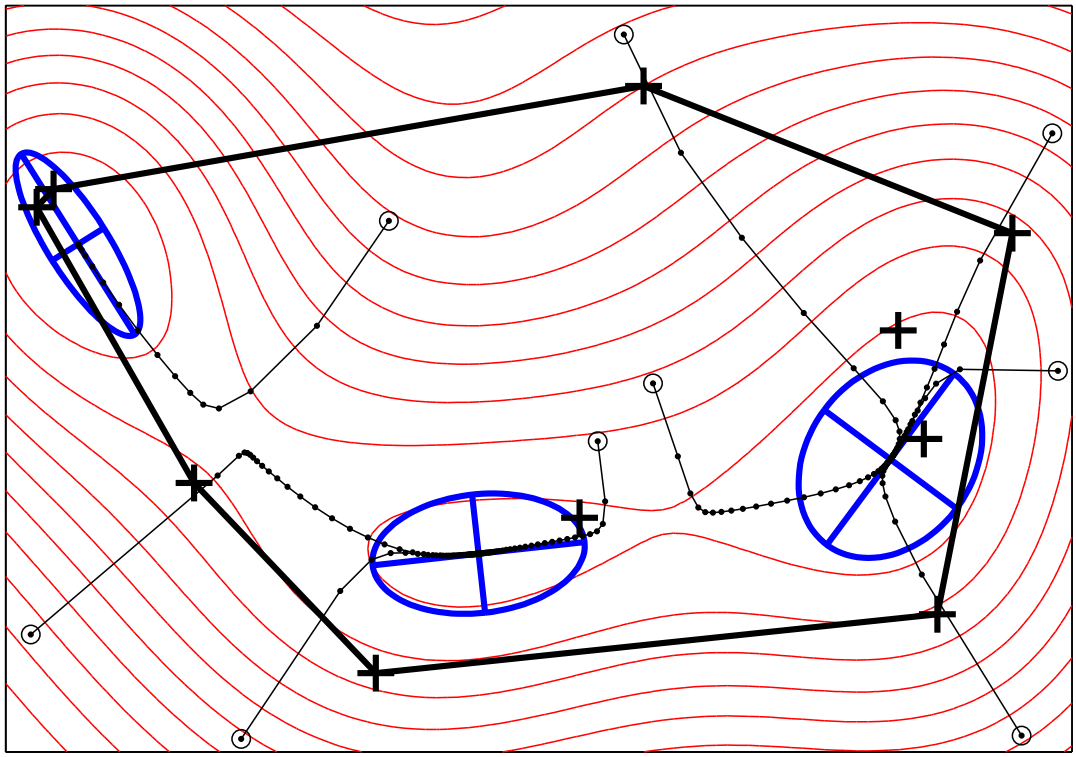

Figure 2: Paths followed by the Gaussian mean shift algorithm for various starting points, overlaid on a contour plot of the Gaussian kernel density estimate $p(\mathbf{x})$ (in the homoscedastic, isotropic case but with non-uniform π_m). The data points $\boldsymbol{\mu}_m$ are marked "+". A mode is located at the centre of each ellipse; the ellipse indicates the eigenvectors (rescaled to improve visibility) of the Jacobian $\mathbf{J}(\mathbf{x}^*)$ at that mode. The thick-line polygon is the convex hull of the data points. Note: (1) all the iterates (except perhaps the initial iterate) and the modes lie inside the convex hull; (2) the paths follow the leading eigenvector when approaching a mode; (3) the EM step size is not the best step size (i.e., does not maximise $p(\mathbf{x})$) along its chosen direction; (4) the EM steps are longer where the gradient is large, but become very small where the density is flat; (5) the slow crawl along ridges of the density, as well as the fact that the iterates may be attracted to saddle points, and then deviate towards a mode; (6) the paths are smooth in that consecutive steps always make an angle in $(-\frac{\pi}{2}, \frac{\pi}{2})$.

modes (see [7, 8] and references inside). Fig. 3 illustrates this for non-gaussian kernels. In 2D or higher, even a Gaussian kernel density estimate can have more than M modes. This is true even in the homoscedastic, isotropic case; for example, in [7] it is shown that if we consider 3 Gaussians with width σ and $\pi_m = \frac{1}{3}$ arranged in an equilateral triangle, then for a certain range of σ there will be 4 modes, three near the centres of the Gaussians and one in the triangle centre. However, it seems that such situation is very rare in practice. Similar arguments have been put forward in the scale-space literature to favour the use of Gaussian kernels for image blurring [17]. This may be one reason why, in practice, Gaussian mean-shift produces better clustering results than other kernels [11].

We have investigated empirically the shape of the convergence domains (i.e., the geometric locus of points that converge to each mode) of Gaussian mean-shift for isotropic kernels in 2D. Fig. 4 shows a representative example, obtained by finely discretising the region of interest and running the algorithm for every point in the grid. In general, the convergence domains are nonconvex (unlike a Voronoi tessellation) and can be disconnected and take quite fancy shapes. Typical features include: one domain can be completely included in another; and the sharper a mode is (e.g. for high π_m and low σ_m), the smaller its domain is. A peculiar, frequent feature is the existence, sandwiched between two domains, of a long, extremely thin stripe belonging to a third domain (see upper inset). Another puzzling feature is the fractal-like aspect of the boundary between some domains (see lower two insets): if one zooms into the region around the boundary, progressively thinner stripes from both domains alternate. This is surprising since one would have expected the boundaries between domains to be lines (in 2D) of points that converge to a saddle point. It suggests that the iterated mean-shift mapping $\mathbf{f}$ might show fractal behaviour for some parameter regimes (Arslan et al. [1] also give an example of the EM algorithm to estimate the location parameter of a 1D Student's t distribution of known scale for which some convergence domains consist of an infinite collection of open intervals). These peculiar domains are probably undesirable for clustering; however, their overall effect is very small (seemingly confined to cluster boundaries) and may be removed during postprocessing.

6 Conclusion

We have shown that the mean-shift algorithm (in its general formulation where each data point can have a different weight and a different, full covariance) is an EM algorithm for Gaussian kernels and a generalised EM algorithm for non-gaussian kernels. This implies that mean-shift converges from almost any starting point, monotonically increasing the density value or leaving it unchanged. We have shown that the order of convergence is generally linear (superlinear for very small or very large widths, and sublinear at mode merges), and that the rate of linear convergence can be very slow for practical cases. We have also shown that the convergence domains of Gaussian mean-shift (and thus of EM algorithms) can be disconnected sets and display a fractal structure.

The EM view suggests possibilities for accelerating the convergence of (Gaussian) mean-shift. Two approaches seem best: (1) accelerating the E step (whose cost is linear in the number of data points) and (2) increasing the convergence order. An example of (1) is sparse EM [21], where one takes full E steps infrequently and partial E steps frequently, with guaranteed convergence to a mode of $p(\mathbf{x})$. In a partial E step the posterior probabilities $p(m|\mathbf{x})$ are updated only for a small subset of plausible data points, which can bring large savings when the data set is large (as is the case in image segmentation). Figure 5 shows savings are indeed possible for an appropriate choice of parameters (size of plausible set, rule when to take a full step, etc.). An example of (2) suggested in [3] is to run EM for several iterations, which typically increases the density value considerably, and switch to Newton's method when approaching a mode and attain quadratic convergence. This may be particularly useful for data sets where the data dimension is low compared to the number of data points (as is the case in image segmentation) since the cost of inverting the Hessian is then negligible. The fact that the mean-shift iterates approach a mode along its local principal component suggests predicting the step as a linear combination of previous iterates (Aitken's method); however, this turns out to be equivalent to Newton's method [18, pp. 141ff]. Other accelerated versions of EM have been developed in the statistical literature [18, chapter 4]. A detailed study and cross-comparison of different acceleration techniques for Gaussian mean-shift is in preparation.

Acknowledgements

We are grateful to Chris Williams for comments on the manuscript, and to an anonymous reviewer for pointing out the relation of the Jacobian to the information matrices in appendix B. This work was partially supported by NSF CAREER award IIS–0546857.

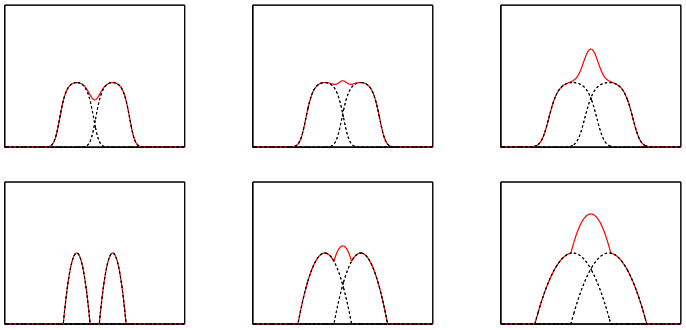

Figure 3: Mixtures of non-Gaussian kernels with more modes than components in 1D. Here, a 1D mixture $p(x) = \frac{1}{2}K\left(\left(\frac{x-\mu_1}{\sigma}\right)^2\right) + \frac{1}{2}K\left(\left(\frac{x-\mu_2}{\sigma}\right)^2\right)$ of two identical kernels K can have from one to three modes. We give two examples: top row, kernel $K(x) \propto \frac{1}{1+e^x/10}$, with infinite support; bottom row, Epanechnikov kernel, with finite support. The thick line represents the mixture and the dashed lines the two individual components. σ increases left to right. In 1D, only Gaussian mixtures have the property of having at most as many modes as components.

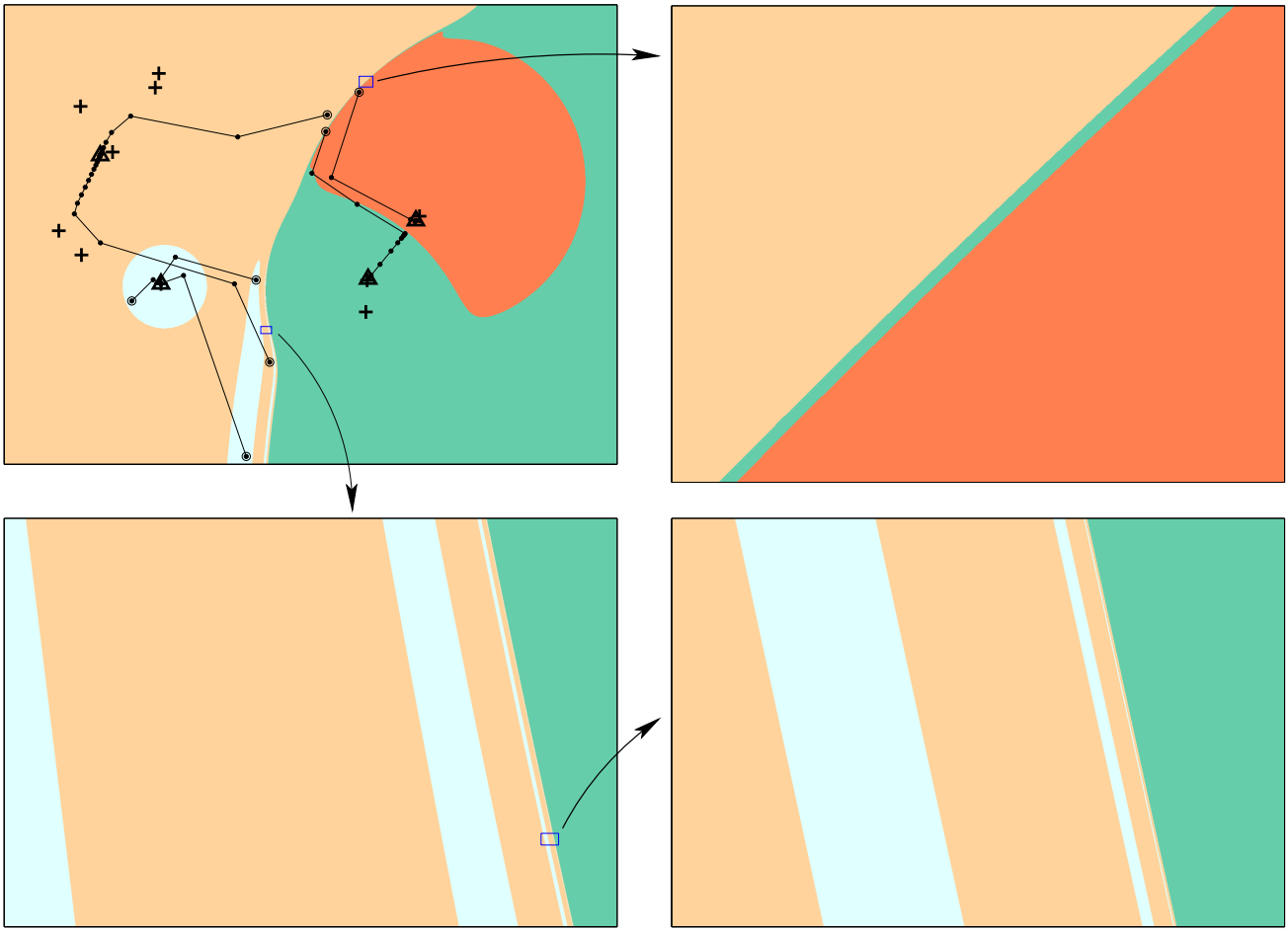

Figure 4: Convergence domains of each mode for Gaussian mean-shift. The figure corresponds to an example with $M = 10$ points in 2D with isotropic covariances (π_m and σ_m are different for each point $\boldsymbol{\mu}_m$) having 4 modes. *Top left*: the 4 convergence domains, i.e., colour-coded plot of $\mathbf{f}^\infty = \mathbf{f} \circ \mathbf{f} \circ \ldots$ The mixture modes are marked "△" and the mixture centroids "+". The search paths for some points are drawn. Note how the domains are generally non-convex sets and can be disconnected. *Top right*: a blowup showing a very thin stripe separating two domains; this feature occurs often with Gaussian mean-shift. *Bottom*: two successive blowups showing a fractal-like pattern in the domains. As one zooms in, thinner and thinner stripes alternate, perhaps indefinitely so (the alternation of stripes continues up to machine precision).

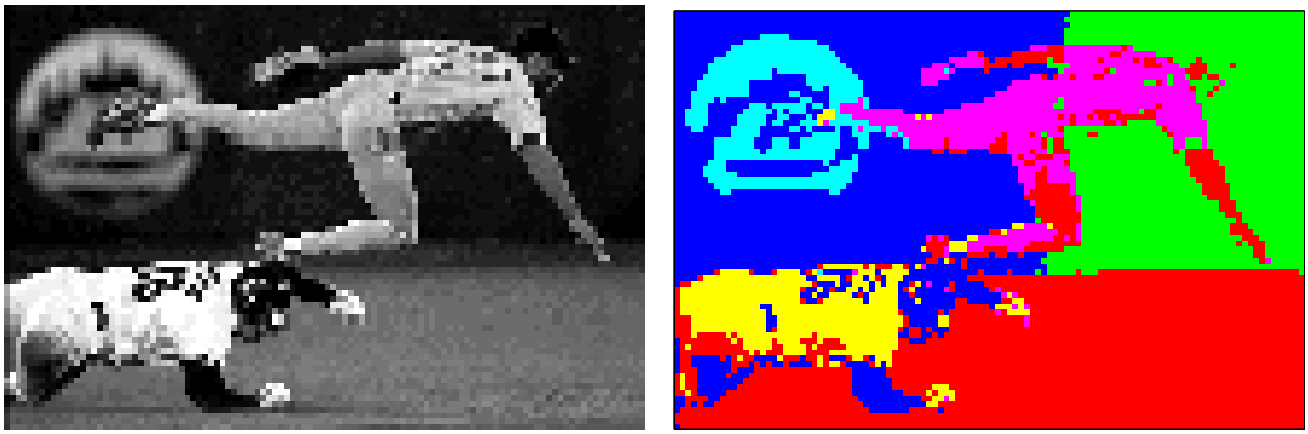

Figure 5: Segmentation of a 110×73 greyscale image with Gaussian mean-shift. The dataset, as usual in mean-shift image segmentation [11], consists of one feature vector (i, j, I) per pixel, where (i, j) is the pixel location in the image and I its greyscale value (normalised to have a similar range to that of i and j). Thus there are $M = 8\,030$ points in $D = 3$ dimensions. Gaussian mean-shift with $\sigma = 12$ and stopping when $\left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^{(\tau)}\right\| < 10^{-3}$ produced the segmentation shown (having 6 clusters) and took 534 620 iterations (i.e., 66.6 iterations per pixel). We modified Gaussian mean-shift according to the sparse EM algorithm [21]. Partial steps update a fixed subset of the M posterior probabilities (the plausible set) containing $0.3M$ of them, and are run till $\left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^{(\tau)}\right\| < 10^{-3}$, or till 20 consecutive partial steps have been taken. Then a full step is run which updates all M posterior probabilities and resets the plausible set to the $0.3M$ nearest neighbours of $\mathbf{x}^{(\tau)}$. The algorithm stops, as before, when $\left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^{(\tau)}\right\| < 10^{-3}$ after a full step. A partial step costs $0.3\times$ as much as a Gaussian mean-shift iteration and we take a full step to cost $2\times$ as much (to account roughly for the extra cost of finding the nearest neighbours). Sparse EM achieved almost the same segmentation (only 3 pixels were misclustered) and took 37 945 full and 541 414 partial steps (respectively, 4.7 and 67.4 per pixel), or equivalently 238 314.2 Gaussian mean-shift iterations (29.7 per pixel). Thus sparse EM was 2.2 times faster than Gaussian mean-shift for this case.

A Dynamical systems view of Gaussian mean-shift

If we consider the iteration index as a continuous variable, the fixed-point iterative algorithm of eq. (5) can also be written as a dynamical system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) - \mathbf{x} = \left(\sum_{m=1}^{M} p(m|\mathbf{x})\boldsymbol{\Sigma}_m^{-1}\right)^{-1} \nabla \log p(\mathbf{x}) \tag{19}$$

where the dot above a variable will indicate differentiation with respect to the time τ, i.e., $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{d\tau}$. Thus, $\mathbf{x}(\tau)$ traces a continuous path in $\mathbb{R}^D$. For the case where $\boldsymbol{\Sigma}_m = \sigma^2\mathbf{I}$ for all m this becomes

$$\dot{\mathbf{x}} = \nabla\big(\sigma^2 \log p(\mathbf{x})\big). \tag{20}$$

Here, starting at a point $\mathbf{x}$, we follow the flux line given by the gradient of $\log p(\mathbf{x})$ (i.e., we perform gradient descent with infinitesimal steps). We can show convergence to a fixed point as follows. Since maximising $p(\mathbf{x})$ is equivalent to minimising $-\sigma^2 \log p(\mathbf{x})$, we can define a Lyapunov function [31]

$$V(\mathbf{x}) = \sigma^2 \log \frac{p(\mathbf{x}^*)}{p(\mathbf{x})}$$

in an open neighbourhood U of every minimum $\mathbf{x}^*$ of $-\sigma^2 \log p(\mathbf{x})$ (i.e., every fixed point of $\mathbf{f}$). V verifies:

1. $V(\mathbf{x}^*) = 0$ and $V(\mathbf{x}) > 0\ \forall \mathbf{x} \in U \setminus \{\mathbf{x}^*\}$, i.e., V is positive definite in $U \setminus \{\mathbf{x}^*\}$.
2. $\dot{V} = \sum_{d=1}^{D} \frac{\partial V}{\partial x_d}\dot{x}_d = \dot{\mathbf{x}}^T \nabla V = \nabla\big(\sigma^2 \log p(\mathbf{x})\big)\big(-\nabla\big(\sigma^2 \log p(\mathbf{x})\big)\big) < 0\ \forall \mathbf{x} \in U \setminus \{\mathbf{x}^*\}$ and equal to 0 at $\mathbf{x}^*$.

So, for the dynamical system of eq. (20), V is a strict Lyapunov function and $\mathbf{x}^*$ is an asymptotically stable point. Thus, the dynamical system converges from any starting point $\mathbf{x}$ in the neighbourhood U to the fixed point $\mathbf{x}^*$ (i.e., it has no cycles or chaotic behaviour). Besides, the convergence near the fixed point is of linear order.

Unfortunately, finding a Lyapunov function for the general case (19) is more difficult.

B Rate of convergence and ratio of information matrices

Here we show that the Jacobian of eq. (16), which determines the rate of linear convergence, can be obtained in terms of the ratio of missing to observed information. Consider the general context of the EM algorithm and let $\mathbf{x}$ represent the observed data, $\mathbf{z}$ the missing data and $\boldsymbol{\theta}$ the model parameters. Define the following functions:

$$H(\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathrm{E}_{p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta}')}\left\{\log p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})\right\}$$
$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathrm{E}_{p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta}')}\left\{\log p(\mathbf{z},\mathbf{x}|\boldsymbol{\theta})\right\} = H(\boldsymbol{\theta}|\boldsymbol{\theta}') + \log p(\mathbf{x}|\boldsymbol{\theta}).$$

Q is the same function as in eq. (9) and represents the expected complete-data log-likelihood given the observed data; H is the expected missing-data log-likelihood given the observed data; and $\log p(\mathbf{x}|\boldsymbol{\theta})$ is the (observed data) log-likelihood. Then the matrices $\nabla^2_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ and $\nabla^2_{\boldsymbol{\theta}} H(\boldsymbol{\theta}|\boldsymbol{\theta}')$ contain the expected, or Fisher, information about $\boldsymbol{\theta}$ in the complete (= observed + missing) and missing data, respectively. Dempster et al. [14, theorem 4] (see also p. 106ff in [18]) showed that, at a convergence point $\boldsymbol{\theta}^*$ of EM, the Jacobian is given by the information ratio

$$\mathbf{J}(\boldsymbol{\theta}^*) = \left(\nabla^2_{\boldsymbol{\theta}} H(\boldsymbol{\theta}|\boldsymbol{\theta}^*)\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\right)\left(\nabla^2_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\right)^{-1}. \tag{21}$$

If we take $\mathbf{z} \to m$, $\mathbf{x} \to \{\mathbf{0}\}$ and $\boldsymbol{\theta} \to \mathbf{x}$ to conform to the notation of section 4, and noting that $\nabla^2 \log p = -\frac{1}{p^2}\nabla p \nabla p^T + \frac{1}{p}\nabla^2 p = \frac{1}{p}\nabla^2 p$ at a stationary point, it is straightforward to show that eq. (21) leads to the same relation of eq. (17). In the interpretation of eq. (21), fast convergence to a mode is obtained when the Jacobian at that mode is small, thus when the proportion of missing data is small. As we saw in section 4, this occurs when $\sigma \to 0$ or $\sigma \to \infty$.

References

[1] O. Arslan, P. D. L. Constable, and J. T. Kent. Domains of convergence for the EM algorithm—a cautionary tale in a location estimation problem. *Statistics and Computing*, 3(3):103–108, Sept. 1993.

[2] O. Bermond and J.-F. Cardoso. Approximate likelihood for noisy mixtures. In *Proc. First Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA'99)*, pages 325–330, Aussois, France, Jan. 1999.

[3] M. Á. Carreira-Perpiñán. Mode-finding for mixtures of Gaussian distributions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1318–1323, Nov. 2000.

[4] M. Á. Carreira-Perpiñán. Reconstruction of sequential data with probabilistic models and continuity constraints. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 12, pages 414–420. MIT Press, Cambridge, MA, 2000.

[5] M. Á. Carreira-Perpiñán. *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*. PhD thesis, Dept. of Computer Science, University of Sheffield, UK, 2001. Available online at `http://www.csee.ogi.edu/~miguel/papers/phd-thesis.html`.

[6] M. Á. Carreira-Perpiñán. Fast nonparametric clustering with Gaussian blurring mean-shift. In *Proc. of the 23rd Int. Conf. Machine Learning (ICML–06)*, Pittsburgh, PA, June 25–29 2006.

[7] M. Á. Carreira-Perpiñán and C. K. I. Williams. An isotropic Gaussian mixture can have more modes than components. Technical Report EDI–INF–RR–0185, School of Informatics, University of Edinburgh, Dec. 2003. Available online at `http://www.informatics.ed.ac.uk/publications/report/0185.html`.

[8] M. Á. Carreira-Perpiñán and C. K. I. Williams. On the number of modes of a Gaussian mixture. In L. Griffin and M. Lillholm, editors, *Scale Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 625–640. Springer-Verlag, 2003.

[9] S. V. Chakravarthy and J. Ghosh. Scale-based clustering using the radial basis function network. *IEEE Trans. Neural Networks*, 7(5):1250–1261, Sept. 1996.

[10] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):790–799, Aug. 1995.

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

[12] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[13] D. DeMenthon. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Statistical Methods in Video Processing Workshop (SMVP 2002)*, Copenhagen, Denmark, June 1–2 2002.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, B*, 39(1):1–38, 1977.

[15] M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(3):471–474, Mar. 2005.

[16] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. Information Theory*, IT–21(1):32–40, Jan. 1975.

[17] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1994.

[18] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, London, Sydney, 1997.

[19] X.-L. Meng and D. B. Rubin. On the global and componentwise rates of convergence of the EM algorithm. *Linear Algebra Appl.*, 199:413–425, Mar. 1 1994.

[20] J. Milnor. *Morse Theory*. Annals of Mathematics Studies. Princeton University Press, Princeton, NJ, 1963.

[21] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1998.

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

[23] K. B. Petersen, O. Winther, and L. K. Hansen. On the slow convergence of EM and VBEM in low-noise linear models. *Neural Computation*, 17(9):1921–1926, Sept. 2005.

[24] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, Apr. 1984.

[25] S. J. Roberts. Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2): 261–272, Feb. 1997.

[26] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. IEEE*, 86(11):2210–2239, Nov. 1998.

[27] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Trans. Neural Networks*, 10(5):1000–1017, Sept. 1999.

[28] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, New York, 1986.

[29] J. R. Thompson and R. A. Tapia. *Nonparametric Function Estimation, Modeling, and Simulation*. SIAM Publ., Philadelphia, 1990.

[30] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conf. Computer Vision (ECCV'04)*, pages 238–249, Copenhagen, Denmark, May 28–31 2004.

[31] S. R. Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer-Verlag, New York, 1990.

[32] R. Wilson and M. Spann. A new approach to clustering. *Pattern Recognition*, 23(12):1413–1425, 1990.

[33] Y. Wong. Clustering data by melting. *Neural Computation*, 5(1):89–104, Jan. 1993.

[34] L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 8(1):129–151, Jan. 1996.