MEAN SHIFT SPECTRAL CLUSTERING

Umut Ozertem[1], Deniz Erdogmus[1], Robert Jenssen[2]

[1] CSEE Department, Oregon Health & Science University, Portland, Oregon, USA

[2] Department of Physics, University of Tromsø, Tromsø, Norway

Abstract. In recent years there has been a growing interest in clustering methods stemming from the spectral decomposition of the data affinity matrix, which are shown to present good results on a wide variety of situations. However, a complete theoretical understanding of these methods in terms of data distributions is not yet well understood. In this paper, we propose a spectral clustering based mode merging method for mean shift as a theoretically well-founded approach that enables a probabilistic interpretation of affinity based clustering through kernel density estimation. This connection also allows principled kernel optimization and enables the use of anisotropic variable-size kernels to match local data structures. We demonstrate the proposed algorithm's performance on image segmentation applications and compare its clustering results with the well-known Mean Shift and Normalized Cut algorithms.

KEYWORDS: Similarity based clustering, Nonparametric density estimation, Mean shift, Connected components, Spectral clustering

1. INTRODUCTION

Clustering has a wide range of applications on several aspects of unsupervised learning; hence, it is a fundamental problem in machine learning. Applications include image segmentation, data mining, data compression, and speech recognition; to name a few. In recent years, a number of authors suggested clustering methods based on the eigendecomposition of a suitable affinity

matrix. Such methods are known as spectral clustering and are considered to be among the most effective methods in the literature. There are several matrix and affinity measures that lead to different spectral clustering algorithms [8,11,16]. The affinity measures that characterize the similarities do not even have to obey the metric axioms except the symmetry property.

Spectral clustering is conceptualized with the use of the second smallest eigenvector of the Laplacian matrix to bi-partition the data [1]. Recently, a number of related clustering methods are suggested that are related to the use eigenvectors or generalized eigenvectors of the affinity matrix. The majority of the spectral clustering algorithms can be interpreted as some variant of graph cut methods [2,3,4], where multiway cuts have also been investigated [5,6]. In addition to these, studies related to the spectral methods are presented in [7,8,9,10,11,14]. Spectral methods are sensitive to the definition of the affinity measure, and choosing a suitable affinity measure is central to this approach. Since no theoretical criterion for choosing the functions to assign the affinities is present in the literature, these algorithms require the assumption of the existence of a suitable affinity definition. Typically, Mercer kernels are utilized as affinity measures, such as the widely used Gaussian kernel.

A different track in spectral clustering was designated by Scott and Longuet-Higgins [12] and later improved by Ng and colleagues [13], where they propose a mapping that uses the eigenvectors of the affinity matrix to transform the data from the original data space to the kernel induced feature space (KIFS), and the actual clustering is performed on the projection of the data in that space. Normalization of the transformed data is an important step in this approach, and clustering of the projected data in the KIFS was shown to be generating very successful results for a variety of different data sets. In this approach, spectral clustering problem becomes a technique for measuring data similarities by an inner product defined in the KIFS. For any Mercer kernel, the

kernel trick defines a technique to compute inner products in the potentially infinite dimensional KIFS. This transformation relies on the assumption that the clustering in the KIFS is easier than in the original data space. In practice, however, this assumption does not hold for all Mercer kernels, and one should search for an optimal kernel design that satisfies this property.

Kernel optimization is known to be a tedious task, and it remains unsolved to the satisfaction of the machine learning community since there are no general and practical propositions in the literature. Furthermore, typically a single kernel does not describe data affinities consistently throughout the whole sample and multiple kernel widths have been used heuristically. To determine a suitable kernel, we use the connection of similarity based kernel methods with kernel density estimation to utilize results from the nonparametric density estimation literature [16].

Mean shift is an iterative nonparametric clustering approach introduced by Fukunaga and Hostetler [15]. This procedure is used for seeking the modes of a probability density function represented by a finite set of samples. Mean shift formulation is revisited by Cheng [17], which made its potential uses in clustering and global optimization more noticeable, and the mean shift algorithm gained popularity [18,19]. Independently, a similar fixed-point algorithm for finding the modes of a Gaussian mixture was proposed and mean shift was shown to be equivalent to expectation maximization (EM) [20,21].

Spectral clustering algorithms require the computation of the eigenvectors of the $N \times N$ affinity matrix, where N is the number of samples. The computational complexity of the eigenvector calculation is $O(N^2)$ per eigenvector, which makes them impractical to use for very large data sets. Typically, by assuming kernels with finite support, the affinity matrices can be made sparse in order to employ efficient techniques such as the Lanczos method.

We propose a mode affinity based clustering algorithm stemming from a variable-size kernel density estimate of the underlying data distribution, which motivates a mean shift like algorithm to represent the data in a much smaller affinity matrix whose size depends on the number of modes of the density estimate. Throughout the paper, we refer to data samples attracted by the same mode in mean shift algorithm as *partition*. We form the affinity matrix between partitions by evaluating a suitable density distance measure and can be processed by standard spectral clustering techniques to determine the final clustering solution. The computational complexity of the second step is negligible compared to other spectral techniques, since the number of modes is much less than the number of samples. The bottleneck is the mean-shift iterations, for which simplifying propositions are discussed. The proposed method is well founded on nonparametric density estimation theory, and the resulting clustering approximates the nonparametric maximum likelihood solution.

2. MEAN SHIFT SPECTRAL CLUSTERING

In this section the details of the proposed method will be discussed. First, we present a brief overview of spectral clustering and then the mean shift in the context of kernel density estimation.

Spectral Clustering: Given a set of data vectors $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ and a suitable kernel function $K(\mathbf{x}_i,\mathbf{x}_j)$ to measure the affinities, the affinity matrix $\mathbf{K}$ and the normalized Laplacian matrix $\mathbf{L}$ are

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) \ , \quad \mathbf{L}_{ij} = D_i^{-1/2}\mathbf{K}_{ij}D_j^{-1/2} \tag{1}$$

where D_i is the normalization term given by

$$D_i = \sum_j \mathbf{K}_{ij} \ . \tag{2}$$

There are a number of different approaches based on the eigendecomposition of either one of $\mathbf{K}$ and $\mathbf{L}$ matrices. Due to the improved eigenspread it provides, $\mathbf{L}$ is the usual choice [8]. Some of these approaches are:

1.Threshold the largest eigenvector of **K** [4].

2.Threshold the second smallest eigenvector of **L** [3].

3.Transform the data to the KIFS using the eigenvectors of **K** or **L** and use a simple clustering algorithm in that domain [12].

Mean Shift Algorithm: The mean shift algorithm is a mode detection procedure based on the density gradient estimation of the data. Given the data set and a kernel function $K_\sigma(.,.)$, where σ denotes the kernel size, the kernel density estimate (KDE) becomes

$$p(\mathbf{x})=(1/N)\sum_{i=1}^{N} K_{\sigma_i}(\mathbf{x}-\mathbf{x}_i) \tag{3}$$

In general, the kernel size could take a different full covariance form for each sample. We experiment with different choices in our simulations. Using (3), the gradient of the probability density of the data is estimated and the local maxima points $\mathbf{y}_c$ are obtained. At these points, the gradient becomes null and the Hessian is negative (semi-)definite:

$$\nabla\hat{p}_K(\mathbf{y}_c)=0 \qquad\qquad \nabla^2\hat{p}_K(\mathbf{y}_c)\le 0 \tag{4}$$

The mean shift iterations are simply fixed-point iterations towards these stationary points. The volume that includes only the set of points that converge to the same mode is defined as the *attraction basin* of that mode.

Recently, spectral clustering approaches based on the affinity and Laplacian matrices have been shown to be essentially related to kernel density estimation followed by an assignment for the class labels that minimizes the inter-cluster overlap and cluster entropy [16]. Particularly, considering spectral clustering with fixed size kernel density estimation in this context, one can easily observe that mean shift becomes an optimization problem, where the *angle* between cluster-means in the KIFS is to be maximized.

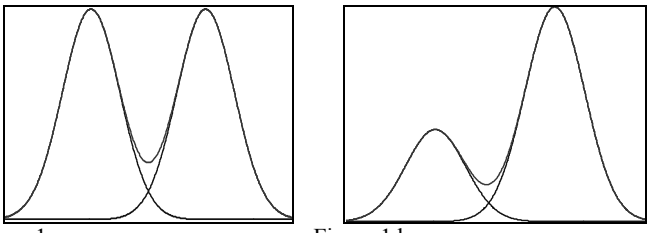

Figure 1.a Figure 1.b

Figure 1. Two Gaussian clusters with (a) balanced and (b) unbalanced a priori probabilities. Dashed lines represent the individual cluster densities, where the Bayes boundary is given by the intersection point. Solid line represents the overall data density, and the approximation to the Bayes boundary is given by the local minimum between the clusters.

Motivated by this relationship between spectral clustering and kernel density estimation we propose a two-step spectral clustering algorithm: the first step determines the modes of the kernel density estimate with a fixed-point iterative procedure in a manner similar to the mean shift procedure. This procedure finds the minimal potential units for clustering, called partitions, which are naturally proposed by the density estimator. The second step employs spectral clustering on a reduced-size affinity matrix consisting of similarities between the M partitions determined in the first step. Typically M is much smaller than N, and this results in significant computational savings in the second step, where the statistically insignificant partitions are merged into significant, larger, and more balanced clusters. In the second step, one can either select a threshold, which automatically determines the number of clusters or one can request the solution for a specific number of clusters using connected components or spectral clustering.

2.1. Decision Boundary for Clustering

In a classification problem the optimal results —in the Bayesian sense— can be obtained by minimizing the Bayes risk function for the given data. For a two-cluster case, this definition of error requires the knowledge of the true underlying class distributions $q_1(\mathbf{x})$ and $q_2(\mathbf{x})$ and their corresponding *a priori* probabilities p_1 and p_2. The corresponding separation boundary is given by the solution to the equation $p_1q_1(\mathbf{x})=p_2q_2(\mathbf{x})$. In a clustering problem, however, we do not have access to the individual class/cluster distributions and the overall data distribution $p(\mathbf{x})$ is known to be $p(\mathbf{x})=p_1q_1(\mathbf{x})+p_2q_2(\mathbf{x})$. The mean shift step inherently determines the boundaries between the

attraction basins of all modes present in $\hat{p}_K$, the kernel-based estimate of this distribution. The importance of modes and saddle points in the context of statistical clustering has also investigated by Comaniciu and coworkers [33]. To illustrate, a one-dimensional scenario is depicted in Figure 1. The local minimum of the overall distribution between the modes is a reasonable approximation to the Bayes boundary. However, two modes that are *supposed* to be in the same cluster can be split artificially by the mean shift algorithm. The merging step that will follow takes care of this shortcoming by combining strongly connected neighboring modes.

2.2.Kernel Density Estimation

In practice, data probability density functions may take complex forms, and determining a suitable parametric family can be a nontrivial task. Nonparametric approaches, on the other hand, overcome this difficulty. The probability density of the data can be estimated nonparametrically using a number of techniques. Techniques based on sample spacing are not differentiable; hence, are not suitable for mean shift iterations [25]. Estimators based on KDE provide a differentiable alternative [22,24]. Furthermore, the effectiveness of KDE in describing arbitrary data distributions are well known [22,24]. KDE may severely break down for accurate density estimation in high dimensional spaces; however, clustering is a simpler problem. Therefore successful clustering still can be achieved with a density estimate that is not acceptable for modeling. In fact, the connection of spectral clustering and mean shift with KDE implies that all of these methods are limited by KDE similarly.

Fixed size KDE allows a natural connection to the spectral clustering methods [16]. On the other hand, variable size KDE is known to have a fast asymptotic convergence behavior [22], which enables us to get a better estimate of the data density with a small number of samples, as well as allowing flexibility to adjust the estimator to local scales in the data distribution. In fact,

the variable size KDE provided better results in our experiments as expected. There is a wide literature on how to select kernel sizes for kernel density estimates, including methods that range from heuristics to principled Bayesian approaches such as maximum likelihood [15,22]. In our experiments, we used several choices for variable and fixed size kernels:

1. Silverman's rule of thumb with spherical Gaussian kernel. For an *n*-dimensional *N*-sample dataset, denoting the sample covariance estimate by $\mathbf{\Sigma_x}$, Silverman's rule gives [25]:

$$\sigma^2 = (1/n)tr(\mathbf{\Sigma}_\mathbf{x})\left(4/\left((2n+1)N\right)\right)^{2/(n+4)} \quad (5)$$

2. Mean of *K* nearest neighbor distances of each sample with spherical Gaussian kernel for variable size KDE times a global scaling factor optimized using maximum likelihood.
3. Covariance of *K* nearest neighbor of each sample with anisotropic Gaussian kernel for variable size KDE times a global scaling factor optimized using maximum likelihood.

The trade-off between clustering performance and computational cost between fixed and variable-size KDE is clear: introducing individual kernel sizes for each sample will increase the overall computational complexity of the algorithm, but will also increase the performance by allowing to obtain a density estimate that is more tuned to local scales and less sensitive to outliers in the data. Silverman's rule is widely accepted to be a suitable choice as fixed-size kernel selection for unimodal distributions; on the other hand, a fixed kernel is not sufficient for densities with complex forms. For variable size KDE, σ_i is selected such that it becomes larger for samples that don't have close neighbors (which are likely to be outliers), leading the probability density function values to be smoother in the vicinity of these samples.

2.3.Mean Shift Iterations

The mean shift algorithm is used to achieve the intermediate clustering results to be used in the spectral analysis step. These partitions of the data distribution provide a natural clustering solution,

where the attraction basin of each mode is a cluster. Given the kernel density estimate of (3), one can design a fixed-point algorithm to map each sample to the mode of the attraction basin that the sample lies in. At the mode, the gradient becomes zero:

$$\frac{\partial p(\mathbf{x})}{\partial \mathbf{x}}^T = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial K_\sigma(\mathbf{x}-\mathbf{x}_i)}{\partial \mathbf{x}} = \mathbf{0} \tag{6}$$

Specifically for a Gaussian kernel (6) becomes

$$\frac{\partial p(\mathbf{x})}{\partial \mathbf{x}}^T = -\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sigma_i^2} G_{\sigma_i}(\mathbf{x}-\mathbf{x}_i)\,(\mathbf{x}-\mathbf{x}_i) = \mathbf{0} \tag{7}$$

Reorganizing the terms in (7), and solving for $\mathbf{x}$ one obtains the fixed-point recursive update

$$\mathbf{x} \leftarrow \left(\sum_{i=1}^{N} G_{\sigma_i}(\mathbf{x}-\mathbf{x}_i)\mathbf{x}_i\right) \Big/ \left(\sum_{i=1}^{N} G_{\sigma_i}(\mathbf{x}-\mathbf{x}_i)\right) \tag{8}$$

The computational load of this phase is $O(N)$ per sample per iteration. In practice, not all samples require the same number of iterations. This step could be made computationally efficient by introducing a stopping criterion to iterate the mean shift procedure starting from each data sample until a satisfactory convergence measure is achieved.[1] Another possibility to reduce computational load is to employ the Fast Gauss Transform [26] or Improved Fast Gauss Transform [29] stemming from the fast multipole concept. Alternatively, a finite support kernel (such as the Epanechnikov kernel) could be employed (as commonly done in image segmentation applications) to limit the number of interactions that need to be evaluated for the update in (8). Combining all of these techniques could result in a fast and efficient implementation of the mean shift phase, with complexity lower than $O(N^2)$. Comaniciu & Meer suggested that using uniform kernels leads to faster convergence of the mean-shift iterations, while Gaussian kernels yield better results [18]. In our experiments, we focus on Gaussian

[1] In particular, we used the following rule: if the K nearest data neighbors of an iterating point does not change for a few iterations, then it must be close to the peak and the neighbors are the data points in the vicinity of the peak.

kernels. Moreover, specific computationally simplifying modifications, such as the use of fast Gauss transform, are discussed later.

In general, one cannot expect each partition to be a *meaningful* cluster mainly due to the existence of statistical variations in nonparametric density estimation in the finite sample case as well as due to the possibility of clusters with multiple modes. Each mode at best represents a vector quantization solution to represent the points in the corresponding attraction basin, which must be evaluated for the final clustering label assignments appropriately to take into account such effects. The method to resolve this issue will be detailed in the next section.

2.4.The Normalized Partition Affinity Matrix

In this section, we propose a similarity measure between partitions of a probability density, which is purely defined in terms of affinities of individual data pairs. In spectral clustering, the data affinity matrix is constructed by evaluating all pair-wise similarities between samples. According to the analogy between kernel affinity measures and kernel density estimation presented in [16], the affinity matrix entry for the *ij*-pair is given by the convolution of the kernels associated with samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in KDE.

Mean shift seeks for the modes of a kernel density estimate. Denoting the underlying *true* probability density function with $p(.)$, mean shift iterations map the data points into the stationary points of $(p * K)(\mathbf{x})$, and the assignment of the data into corresponding modes is done accordingly. Since the results of the mean shift is solely based on the underlying KDE, one should consider the resulting intrinsic data manifold to analyze the modes. In fact, KDE maps all the data points on a spherical manifold in the kernel induced feature space (KIFS), and in order to investigate the intrinsic data manifold in further detail, one should first grasp the characteristics of the data transformation into KIFS.

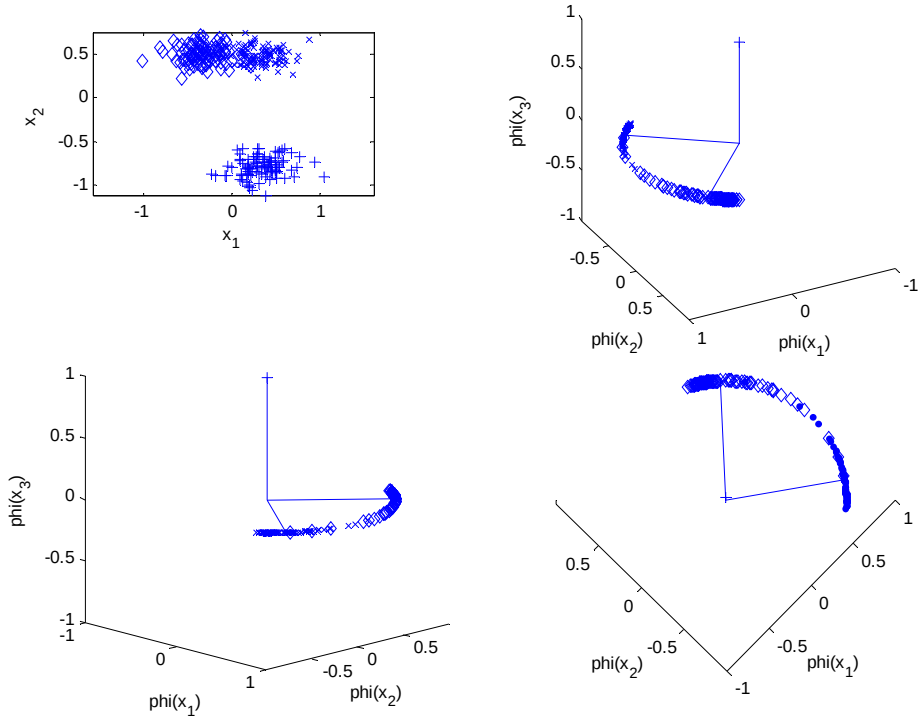

Figure 2. The three modal Gaussian dataset is shown in the original feature domain, where the data points from each mode are shown by "+", "×", and "◊" (top left). The three dimensional representation of the mapping in the kernel induced feature space is shown from different angles using the same markers "+", "×", and "◊" for different modes.

KIFS is a potentially infinite dimensional space spanned by the eigenfunctions of the kernel function[2]. Generally, the eigenvectors of the data affinity matrix are employed to approximate these eigenfunctions and the KIFS is represented in a dimensionality that is equal to the number of data samples. The mapping from the original feature domain into the KIFS is defined by the *kernel trick*. Specifically, for a translation invariant nonnegative[3] kernel function, the mapping into KIFS is defined on the unit hyper-sphere. To investigate this fact deeper in detail, one should explicitly rewrite the kernel function in terms of its eigenvalues and eigenfunctions. For illustrative purposes and simplicity here we start with a fixed kernel function; so, the eigendecomposition of the kernel function becomes

[2] According to the theory of reproducing kernels for Hilbert spaces (RKHS), for every positive semi-definite kernel function $K(.)$ that satisfies the Mercer conditions [30], the set eigenfunctions $\{\varphi_1(\mathbf{x}),\varphi_2(\mathbf{x}),...\}$ form a basis for the Hilbert space of square integrable nonlinear functions [31].

[3] This requirement stems from the connection to the density estimation.

$$K(\mathbf{x}_i - \mathbf{x}_j) = \sum_{k=1}^{\infty} \lambda_k \varphi_k(\mathbf{x}_i)\varphi_k(\mathbf{x}_j) \cong \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\Lambda}\boldsymbol{\varphi}(\mathbf{x}_j) \geq 0 \tag{9}$$

Here, the outcome of the kernel function for a particular data pair can be regarded as the cosine of the angle between them in KIFS domain. Also note that particularly for a nonnegative kernel function the cosine between all the transformed data pairs are less than $\pi/2$; that is, all the data is transformed into a single quadrant in the KIFS. Illustrations regarding to a simple scenario of a three modal data distribution is depicted in Figure 2 for varying degrees of overlap between the modes. Since it is not possible to fully present the neither the infinite dimensional KIFS domain nor its widely used *N*-dimensional approximation, a subspace projection into the span of three greatest eigenvectors is used here. This method is proposed by Ng and colleagues [13] to analyze spectral clustering specifically; however, it can be generalized to be used for any kernel-based method to analyze the mapping into KIFS. The subspace projection of the data in KIFS domain is shown from three different angles, along with the three modal Gaussian mixture data in the original feature domain. The most important point here is to observe that the data points in the mode denoted with "+" is mapped onto the sphere such that its mean is perpendicular to the plane spanned by the class means of the other two modes denoted with "×" and "◊" (top right), and the angle in between the other two mode means are less than $\pi/2$ (bottom right), even with only three leading eigenvectors[4]. However, this analysis is only for illustrative purposes, and one can use the values of the normalized mode affinity matrix to see the actual *angles* in between the partition means of the transformed data in KIFS. A suitable metric for assessing whether two partitions belong to a single cluster is the angle between the means of these two partitions on the infinite dimensional spherical manifold. The angle between partition densities can be written as

[4] Particularly, in this example the ratio of the magnitude square of eigenvalues correspond to the basis of the subspace that used for illustration to the sum of magnitude square of all eigenvalues is 0.67.

$$D_{ij} = \frac{\mu_{C_i}^{\Phi\,T} \mu_{C_j}^{\Phi}}{\sqrt{\left\|\mu_{C_i}^{\Phi}\right\|}\sqrt{\left\|\mu_{C_j}^{\Phi}\right\|}} \tag{10}$$

where $\mu_{C_i}^{\Phi}$ is the mean of the i^{th} partition of the transformed data. Ideally, one should use the intrinsic means[5] on the sphere here; however, assuming a well-chosen kernel that results in compact clusters in the KIFS and for computational simplicity, here we approximate the intrinsic means with the Euclidean mean. As one can observe from Figure 2, the partitions have a low spread on the sphere for a suitable choice of the kernel, and Euclidean distance definition becomes a suitable approximation of the intrinsic mean. Specifically, this characteristic is more obvious for the isolated modes that are far from any other mode, as an example, in all the data points that belong to the mode denoted with "+"looks as if it is mapped into a single point in the KIFS representations in Figure 2. Euclidean distance reduces the computational load and proved to be efficient in the experiments. Substituting the sample averages, one can rewrite (10) as

$$D_{ij} = \frac{\sum_{x_i \in C_k}\sum_{x_j \in C_l} \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\Lambda}\boldsymbol{\varphi}(\mathbf{x}_j)}{\left(\sum_{x_i \in C_k}\sum_{x_j \in C_k} \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\Lambda}\boldsymbol{\varphi}(\mathbf{x}_j)\right)^{1/2}\left(\sum_{x_i \in C_l}\sum_{x_j \in C_l} \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\Lambda}\boldsymbol{\varphi}(\mathbf{x}_j)\right)^{1/2}} \tag{11}$$

To investigate how this translates into the original feature space we rewrite (11) using (9) as

$$D_{ij} = \frac{\frac{1}{N_k N_l}\sum_i\sum_j K(\mathbf{x}_k^i - \mathbf{x}_l^j)}{\left(\frac{1}{N_k^2}\sum_i\sum_j K(\mathbf{x}_k^i - \mathbf{x}_k^j)\right)^{1/2}\left(\frac{1}{N_l^2}\sum_i\sum_j K(\mathbf{x}_l^i - \mathbf{x}_l^j)\right)^{1/2}} \tag{12}$$

where $\mathbf{x}_k^i$ denotes the k^{th} sample associated with mode i. Hence, the affinity measure given in (12) is essentially the sample average of the affinity measures of the individual data samples normalized

[5] The data means obtained using the intrinsic distances on the intrinsic manifold (distances evaluated over the sphere).

over the individual mode *volumes*, where the sample average is calculated over the pairs of samples corresponding to those particular modes. Therefore, translating the mode-pair affinity defined in KIFS into the original feature domain, it is important to note that the distance measure in (10) is the normalized graph cut defined in between these modes. At the limit of infinite samples it converges to the cosine of the *angle* between the two distributions $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$ in the function space according to the Euclidean inner product definition, where the inner product is

$$< p_i(\mathbf{x}), p_j(\mathbf{x}) >= \int p_i(\mathbf{x})\ p_j(\mathbf{x})\ d\mathbf{x} \tag{13}$$

and $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$ are the corresponding probability density functions of the modes. Since we employ mean shift algorithm in the prior step, we inherently use the KDE based estimate of this inner product, which is given by

$$< \hat{p}_i(\mathbf{x}), \hat{p}_j(\mathbf{x}) >= \frac{1}{N_k N_l} \sum_i \sum_j K(\mathbf{x}_k^i - \mathbf{x}_l^j) \tag{14}$$

and the similarity measure (14) is nothing but the angle defined by this inner product. Hence, we can express (12) as

$$D_{ij} = \frac{< \hat{p}_i(\mathbf{x}), \hat{p}_j(\mathbf{x}) >}{< \hat{p}_i(\mathbf{x}), \hat{p}_i(\mathbf{x}) >^{1/2} \quad < \hat{p}_j(\mathbf{x}), \hat{p}_j(\mathbf{x}) >^{1/2}} \tag{15}$$

In order to explore the effects of finite sample sizes on the distance measure based on the inner product estimate the bias and variance of this inner product estimator are investigated in the Appendix. The asymptotic unbiasedness and consistency conditions are the same as those of the kernel density estimators employed.

At this point, to provide consistency with the related literature, we rewrite the derivation provided for the proposed pdf distance measure in matrix form. Since the affinity matrix **K** is also implicitly used in the prior mean shift step, the matrix representation has also advantages in the implementation. The affinity matrix **K** is defined as follows

$$\mathbf{K}_{kl} = K(\mathbf{x}_k - \mathbf{x}_l) \tag{16}$$

Using (15) one can rewrite the summations over the corresponding rows and columns of the kernel matrix and one can build a (normalized) partition affinity matrix $\widetilde{\mathbf{K}}$, whose entries are the pairwise affinities in (13) rewritten using the matrix representation.

$$\widetilde{\mathbf{K}}_{kl} = D_{kl} = \frac{\frac{1}{N_k N_l}\sum_{i\in k}\sum_{j\in l}\mathbf{K}_{ij}}{\left(\frac{1}{N_k^2}\sum_{i\in k}\sum_{j\in k}\mathbf{K}_{ij}\right)^{1/2}\left(\frac{1}{N_l^2}\sum_{i\in l}\sum_{j\in l}\mathbf{K}_{ij}\right)^{1/2}} \tag{17}$$

Note that the procedure applied to define the distances between the modes in KIFS does not depend how the initial clustering assignments have been obtained. Basically, instead of mean shift, one can also use another clustering algorithm here due to computational complexity limitations or requirements for heuristic rules specially designed for the particular dataset. For simplicity in the above illustrations, a fixed-size KDE is used throughout this subsection. However, as stated before, variable size KDE is a more powerful tool for estimating the probability density, and considering the natural connection to affinity based clustering methods, leads to better clustering results.

2.5. Connected Components of the Partition-Graph

Once the partition affinity matrix is evaluated, any standard spectral clustering method such as minimum graph cuts and normalized graph cuts can be employed on matrix of mode-affinity measures given in (13). Determining the eigenvectors of this matrix would cost $O(M^2)$ per eigenvector; negligible compared to the complexity of $O(N^2)$ when applied to the data affinity matrix. To illustrate this, we applied different spectral clustering methods to $\widetilde{\mathbf{K}}$ for different datasets. Along with the well-known methods we also propose utilizing another simple algorithm here based on finding the connected components with a complexity of $O(M^4)$, which would make the algorithm become impractical for large affinity matrices. On the other hand, this method

Table 1. Outline of the overall algorithm.

1. Get the data **x** and select the kernel size (or variable kernel sizes for each data point) using any of the methods given in section 2.2. Note that the computational complexity of these methods for the kernel selection is increasing with increasing index.

2. Employ the fixed-point iteration in (8) to find the partitions of the probability density function.

3. Construct **K**, using (16) calculate D_{ij} for all i,j and using (17) and construct $\widetilde{\mathbf{K}}$.

4. Sort all pairwise affinities defined in non-diagonal entries of $\widetilde{\mathbf{K}}$ in an ascending order. The diagonal entries can be ignored, since they all are equal to unity. Representing the affinities of partitions with themselves, these entries don't carry out information.

5. Remove the weakest connection, defined by the smallest affinity.

6. Check graph connectivity, and determine the number of separate trees. If the number of separate trees in the graph is equal to the required number of clusters, assign the connected partitions into the same cluster and stop. Otherwise go to step 5.

produced good results for small sized mode affinity matrices and is mostly preferred in our experiments. Alternatively, hierarchical clustering could also be used to analyze the mode merging structure.The simple procedure of determining the connected components of the mode graph given the (normalized) affinities basically sorts all affinities in $\widetilde{\mathbf{K}}$ in ascending order and removing the weakest connections determined by the smallest affinity values one by one until a predetermined number of clusters is reached. In each step, the graph connectivity is being checked and the algorithm decides on either continuing to remove connections or stopping and assigning the connected modes into the same cluster. Performed in each iteration with $O(M^2)$ complexity, checking the graph connectivity is the dominant computational load, resulting in a $O(M^4)$ complexity for the overall algorithm. To check the graph connectivity, a well-known connected components algorithm is used [23]. The outline of the resulting algorithm is given in Table 1.

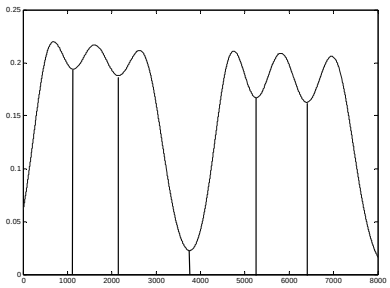

Figure 3. A probability density function obtained by a Gaussian mixture of 6 components. The dashed lines represent the boundaries for each attraction basin.

The procedure described above requires the number of clusters to be preset by the user. Alternatively, one could define a threshold for the (normalized) affinity values between pairs of partitions, which could be employed to maintain the connections corresponding to the larger affinities. Selecting a *good* threshold could be achieved by observing the clustering structure of the data as the threshold is increased from 0 to 1. The clusters that *survive* for a larger interval of threshold values could be deemed statistically significant. This procedure has been previously employed for setting temperature and kernel size in clustering algorithms [27], and is similar to hierarchical clustering in principle.

Constructing $\widetilde{\mathbf{K}}$, one can have an idea of the distances between the partitions of the overall distribution and the partition-affinity analysis step provides a systematic way of merging the modes by statistically investigating the possibility that neighboring partitions might belong to the same cluster. Previously, this issue has been addressed using the Capture Theorem, which requires additional expensive mean-shift iterations to be executed [18]. In the proposed framework, the connectivity of such partitions can be decided using the preset number of clusters approach or the survivability of the clusters along the threshold axis as described. To illustrate this, consider the probability density estimate presented in Figure 3. In this figure, decision boundaries of the standard mean shift algorithm, namely the boundaries of each neighboring attraction basin is given by dotted lines. It can be argued that there are two *statistically significant* clusters in this

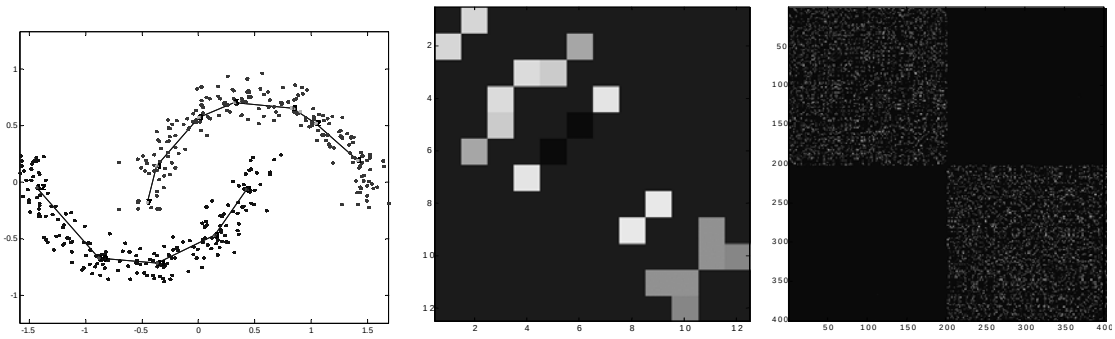

Figure 4.a Figure 4.b Figure 4.c

Figure 4. The crescent dataset is presented in (a), where the dots represent the data points and the lines connect the modes of the estimated distribution after connected components are determined. Normalized mode affinity matrix and the data Laplacian matrix are shown in (b) and (c), respectively (diagonals nulled).

probability density; however mean shift still defines each attraction basin as an individual cluster. Although one can argue that it might be possible to manipulate the kernel size such that a two cluster solution is obtained, the partition-affinity analysis and the connection to kernel density estimation eliminates the requirement for such heuristic attempts or tedious kernel optimization tasks, and provides a principled way of choosing the kernel and merging the obtained partitions.

The problem of deciding about statistically significant clusters is tackled before by Comaniciu and Meer [18], where they use mean shift to obtain intermediate clustering result and propose a connected components based clustering method, which performs a neighborhood search over convergence points, namely the modes, to obtain clustering results. Besides, this approach is representing each partition only with the location of the corresponding peak point of the pdf in the merging step, and does not necessarily optimizes a cost function over the probability densities of the individual modes. We will provide comparison with this approach in the next section.

Overall, the proposed algorithm provides a systematic and nonparametric approach for estimating perceptually and statistically important clusters. This will be illustrated in the next section in artificial data and image segmentation applications.

3. EXPERIMENTAL RESULTS

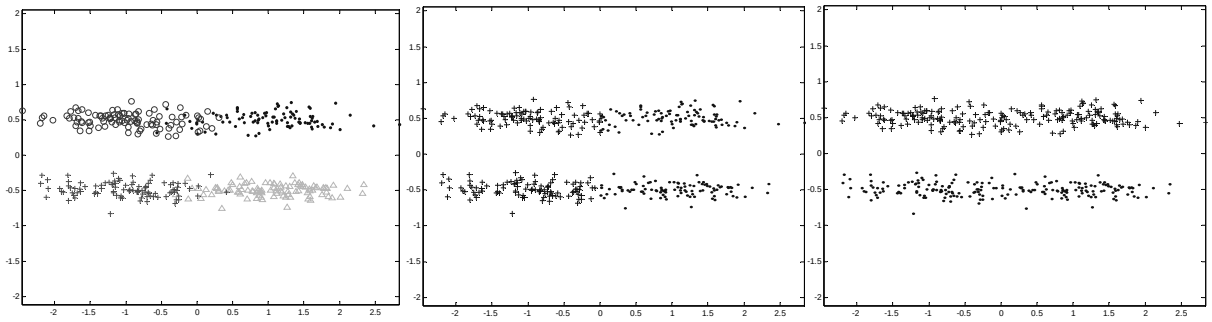

Figure 5.a Figure 5.b Figure 5.c

Figure 5. (a) the four modal Gaussian dataset. (b) clustering results obtained with the earlier mode merging method (c) clustering results for MSSC

Figure 6.a Figure 6.b Figure 6.c

Figure 6. The original image is shown in (a). Clustering results for fixed-size spherical kernel obtained using Silverman's rule and variable-size anisotropic kernel using local neighborhood covariances are presented in (b) and (c), respectively.

The proposed algorithm will be illustrated on a synthetic dataset and on image segmentation using fixed- and variable-size kernel estimates. In all experiments, Gaussian kernels are utilized. For variable-size cases, the local kernel covariance is determined by the sample covariance of k-nearest neighbors and a global scaling parameter.

Crescent dataset: This dataset is synthetically generated and consists of two crescent-shaped clusters with a nonlinear separation boundary. For each cluster, 200 two-dimensional samples are generated by uniformly selecting the angle in a π-radian arc and perturbing the radius with Gaussian distributed random values. A fixed-size spherical kernel determined by Silverman's rule-of-thumb, and the proposed graph connectivity clustering method are used in this simulation. A realization of this dataset and the corresponding clustering results are shown in Figure 4, where the modes detected by the mean shift algorithm are labeled as shown. Connected

modes in the figure represent the clustering result. The data Laplacian matrix and the normalized partition affinity matrix are also shown in Figure 4.

Comparison with an earlier mean shift mode merging method: In this subsection, we will compare our results with those of another widely used mode merging approach proposed by Comaniciu and Meer [18].. Defining the distance measure between partitions, one can either use a preset value for the distance allowed or employ a connected components algorithm and remove the weakest connections until the desired number of clusters is reached. Rephrasing, this earlier mode merging approach can be summarized as follows:

1. Run the mean shift. Store the convergence points and corresponding data assignments.
2. Calculate the distance between all the convergence points and merge the clusters if the distance in between the convergence points is less than a preset threshold.
3. Perform the data assignments over the merged clusters to obtain the final clustering.
4. Remove regions that contain fewer data samples than a preset threshold.[6]

Like all other pairwise similarity based approaches, the most critical point here is the definition of the distance measure. Starting with an illustrative example, results for a four modal Gaussian mixture is compared in Figure 5, where the Gaussian components are centered at $(\pm 1, \pm 1/2)$, each with 100 data samples. Figure 5a shows the dataset, where the data samples belonging to each cluster is denoted with a different symbol, and mode merging results for two output clusters are shown in Figure 5b and Figure 5c. Figure 5b shows the mode merging results for the earlier method that considers pairwise distances between maxima points of modes, and Figure 5c shows the results obtained by the pdf distance estimate that we propose. As expected, results based on pdf distance are more natural clustering solutions for this toy dataset. We will also present comparisons with this algorithm on real data in the quantitative performance evaluation section.

[6] This step can be regarded as specific to image segmentation applications. For details see the original paper [18].

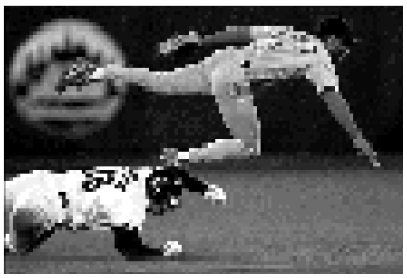

Figure 7.a

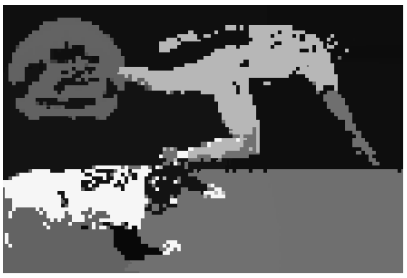

Figure 7.b

Figure 7. The original grayscale baseball player image is presented in (a). Clustering result for the mean-shift step using variable-size anisotropic kernels with local neighborhood covariances is shown in (b).

Note that the threshold in step 2 attempts to identify modes that are close to each other, while the thresholding in step 4 attempts to eliminate modes with small "volume". The angular density distance in (17) achieves both objectives simultaneously and consistently.

Comparison of fixed kernel size and variable kernel size: Image segmentation is an important application of clustering and has been utilized as a benchmark in mean shift and spectral clustering literatures. Depending on the context, a variety of suitable feature vectors that distinguish specific objects and textures from each other and the background can be derived. For simplicity, in this example, we present segmentation results for a color image using pixel coordinates and the RGB channel intensities as the features. The *buddies* image is 96×128, resulting in a huge affinity matrix. In the spectral clustering literature, a number of authors suggested using kernels that are nonzero only in the vicinity of a pixel when defining similarities, in order to reduce computational complexity. This corresponds to kernel density estimation with a finite support kernel along the pixel coordinate features, and relies on the assumption that the points in the same cluster should form a spatially continuous pattern. Note that pixels spatially separated by a distance larger than the kernel extent can still be in the same cluster through local connections.

Results for both variable and fixed kernel sizes are presented in Figure 6. For the fixed-size spherical kernel case, the data is normalized to unit-variance in each dimension to avoid scale-based performance degradation. Hence, this method can in fact be considered to use a separable

kernel with size tuned to the standard deviation of each feature individually. Due to data dimensionality, it is not possible to show the samples in the feature space, and only the input and output images will be presented in grayscale. In the output image, the pixels with the same grayscale value belong to the same cluster. The grayscale values themselves do not represent any information; however, the differences between grayscale levels of the segments represent the distance in the clusters in the feature space as measured by the angular density distance.

Comparison with Normalized Cut: The baseball player image has been utilized previously in spectral clustering papers (e.g., [3,13]) and its segmentation is performed using the proposed algorithm to provide a means of comparison. Features used in the example are the grayscale pixel intensities and the pixel coordinates. The original *baseball player* image, which is 147×221, is shown in Figure 7 with the clustering results obtained using a variable anisotropic kernel size. Fixed kernel size results using Silverman's rule are presented in Figure 8 along with results of the well known normalized cut algorithm proposed in [3]. Normalized cut is a widely accepted algorithm that employs multi-way graph cut for the required number of clusters. It is not considered to be the state of the art for image processing applications, however, among the clustering methods in the literature that only require the number of output clusters but no other parameters, normalized cut is known to be one of the most efficient methods to provide balanced and statistically important clusters. Same as normalized cut, our approach MSSC only requires the number of output clusters. In addition, as described above, the angular distance measure in (12) is related to the normalized cut cost; hence, the comparison is logical at this point. In Figure 8 results of mean shift spectral clustering and normalized cut are presented for different number of clusters. We observe that the mean shift spectral clustering algorithm generates perceptually more relevant clusters even at small target number of clusters.


Figure 8.a

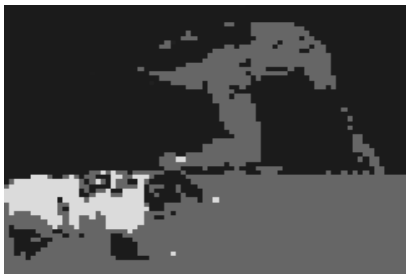
Figure 8.b


Figure 8.c

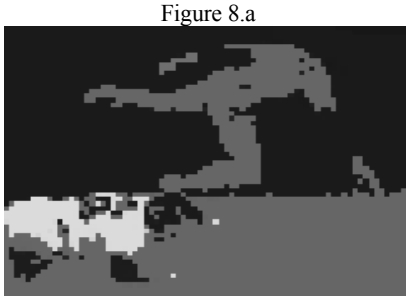
Figure 8.d

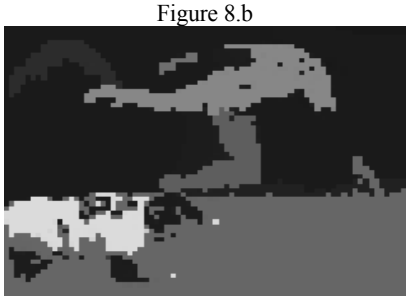
Figure 8.e

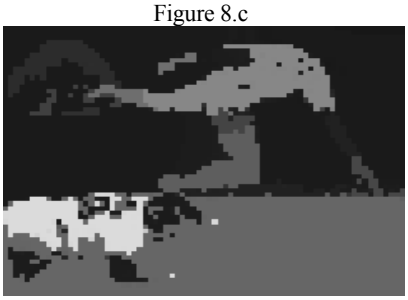
Figure 8.f

Figure 8. Results of the Normalized Cut algorithm for 5, 10, and 15 clusters are presented in (a), (b), and (c). Results of the Mean Shift Spectral Clustering are presented in (d), (e), and (f) for the same number of clusters, respectively.

Quantitative Performance Evaluation: Due to lack of the definition of the *correct* or *optimal* segmentation boundary, in all preceding experiments, we left the performance evaluation of the results to the reader's visual inspection. In this subsection, we utilize the Berkeley Segmentation Dataset and Benchmark [32] to present a quantitative performance evaluation of mean shift spectral clustering. This dataset has also the corresponding human segmentation labels along with the images that it contains, and provides an empirical basis for image segmentation algorithms. In the evaluation, we used one of the human segmentations as the ground truth for the segmentation boundary, and to be able to overcome the difficulty of inconsistencies among human segmentations, we performed our experiments on some images or some parts of images, whose segmentations corresponding to different human subjects were similar.

For performance evaluation we used region based precision-recall method. To obtain the precision recall curve, we compute two quantities, precision and recall, for different values of the kernel size. Although precision-recall curve resembles the ROC curve for hypothesis testing it is fundamentally different, and it is widely accepted as a powerful quantitative summary for

evaluating performance in the image segmentation literature. Precision (P) is defined as the ratio of detections that are true positives rather than false positives and recall (R) is the ratio of true positives rather than the ones that are missed. Since the both precision and recall is defined for a two class scenario, probability of detection and miss is meaningless for a multiple cluster case. To transform the multiple cluster segmentation problem into a detection problem, we build a segmentation matrix S both for the obtained segmentation result and the ground truth, such that S_{ij}=1 if i^{th} and j^{th} data samples belong to the same segment of the image, and zero otherwise. Having the segmentation matrices constructed by using the ground truth and the results obtained using the segmentation algorithm, precision and recall values for can simply be obtained for the corresponding kernel size.

Although the precision-recall curve fully represents the performance characteristics for different parameter values, it is useful to generalize the performance with a single number, and the *F-measure* is used here for this purpose. The *F-measure* is defined as weighted harmonic mean of precision and recall and given as

$$F = 1/\alpha P^{-1} + (1-\alpha)R^{-1} \tag{18}$$

The position of the maximum *F-measure* along the precision-recall curve defines the optimal segmentation performance for the given particular α value; hence, the maximum *F-measure* along the precision-recall curve can be used to summarize the performance. The weighting factor α is set to 0.5 in our experiments.

Figure 9 presents the output of mean shift and mode merging results both for the earlier method proposed by Comaniciu and colleagues and mean shift spectral clustering along with the original images used in the experiments. While generating the mode merging results for these two algorithms we use the same mean shift output, and in the mean shift iterations we used a variable

(a) (b) (c)

Figure 10. (a) an eight-cluster mean shift results with a wide kernel (b) mean shift results for Silverman's rule (c) mean shift spectral clustering solution for an eight-cluster solution

kernel size for the reasons we mentioned earlier. Particularly, the covariance of the Gaussian kernel function used in this experiment is given by $\Sigma_i = \sigma^2 C_i$, where C_i is the covariance of the K-nearest data points, and σ^2 is a global scale parameter. Since it leads to a biased density estimate, choosing a fixed number of data points to use for local covariance estimation is problematic from the density estimation point of view[7].

We set this value as $K = \sqrt{N}$ and obtain the precision-recall curve for different values of the global scale σ^2. We use (x,y) coordinates of the pixels and the intensity value I(x,y) as features in this experiment. The kernel function used for the mean shift entirely defines the distance measure for MSSC; on the other hand, since no optimization method is proposed in the earlier mode merging method to select the threshold that defines the neighborhood of modes that should be connected, we performed a brute-force search for different threshold values and present the results with the best F-measure among those correspond to the same mean shift output for the particular kernel size.

Comparison with Mean Shift: Instead of selecting the kernel sizes stemming from the density estimation literature, one may try to select the kernel size that leads to desired clustering results. Most common way to do this is to run the algorithm several times for different kernel sizes, until a performance measure is optimized. For example, recalling the density estimate given in Figure 3,

[7] Generally, to obtain an asymptotically unbiased density estimate K should satisfy: $\lim_{N\to\infty} K = \infty$, $\lim_{N\to\infty} K/N = 0$

one may argue that it is possible to select a wider kernel size to obtain a bimodal density estimate for the same dataset. In image segmentation literature, the rule of thumb for an $N \times N$ is given as: (i) scale the intensity level to *N*, so that the data is normalized among the pixel coordinate and intensity features (ii) select the kernel size as *N*/5. Here, the selection is rather heuristic, but it is reported to lead good image segmentation results [34]. Running mean shift with a wide kernel is a widely used approach along with the mode merging method that we presented performance comparisons in the previous subsection. In this subsection, we compare our approach with the mean shift itself. Note that the mean shift clustering results obtained here are totally different than the intermediate mean shift results in MSSC. In the context of mode merging, the mean shift results are required to have many more clusters as compared to the final number of clusters, for this reason either relatively narrow Gaussian kernels or finite support kernels such as the Epanechnikov kernels are used. For the results that we obtained for the mean shift algorithm, here we used Gaussian kernels and utilized the rule of thumb given above to find the kernel size. For mean shift spectral clustering we used Silverman's rule to obtain the results. Figure 10 shows the results of these two methods on the baseball player image. For the image normalized along the feature dimensions as described above, Figure 10a shows the results of the mean shift algorithm, where mean shift ends up with an eight cluster solution. If the kernel size is selected according to Silverman's rule given in (5), mean shift ends up with the clustering solution shown in Figure 10b, and Figure 10c shows the eight cluster mean shift spectral clustering solution obtained using this mean shift results. Comparing the Figure 10a and Figure 10c one can see that mean shift spectral clustering results in a better segmentation, whereas in the mean shift results the body parts of the upper baseball player are clustered together with the ground, and the ground and background wall are artificially split into two clusters.

4. CONCLUSIONS

Although proven to be effective and considered to be among the state-of-the-art methods for clustering, the main disadvantage of mean shift is that the resulting clustering assignments can artificially separate two modes that are supposed to be in the same cluster. The proposed pdf distance based mode merging stage allows a theoretical connection between the *quality* of the clustering solution and that of the kernel density estimate. One can exploit the rich literature on kernel optimization based on density estimation performance, thus eliminate the expensive need for kernel size selection via repeated clustering trials. This also allows a natural introduction of the stronger variable-size anisotropic kernel density estimates to spectral clustering algorithms.

Merging clusters obtained by mean shift is a well-known problem, and it was handled before by considering the distances in between pairs of the mode peaks. As compared to this earlier approach, MSSC provided slightly better results for image segmentation applications. However, the only point here is not the improvement in the performance itself. Since the distance measure defined here is explicitly a function of the density estimate implied by the mean shift, MSSC does not require another parameter to evaluate the mode distances and eliminates the necessity of a parameter search or optimization step here. The complete process is automated except the choice of number of clusters. At this point, one may argue that the computational complexity that (17) requires $O(N^2)$ computations, as compared to more inexpensive $O(M^2)$ complexity [18]. However, note that the pairwise kernels that are required in (17) have already been evaluated in the preceding mean shift step. Hence, (17) does not introduce any significant computational load, but one needs to save the pairwise similarity matrix in the first iteration of the mean shift.

For image segmentation applications, mode distances based method is able to provide satisfactory performance, most probably due to the characteristics of the commonly used feature space; x, and y coordinates, and the pixel intensities - the coordinate values generate evenly spaced samples in the feature space. On the other hand, using distances between peaks corresponds to modeling the mode just with the location of the peak, which is usually insufficient, and may lead to some unnatural clustering solutions for a general clustering problem. This drawback of the earlier approach is demonstrated with a simple illustration, where the Gaussian distributions are not spherical. But note that, the mode affinity that MSSC defines is able to model the linear or nonlinear shapes of the modes. In image segmentation literature, there are two main ways of using mean shift: (i) use a narrow kernel and merge the modes using peak distances (ii) use wide kernels and run mean shift several times until a performance measure is satisfied. Here, we compare our results with both of these approaches, and we obtain similar or better performances with no effort spent on parameter tuning.

While the mean shift procedure in its raw form has $O(N^2)$ complexity, this load can be reduced by clever choice of kernel supports, such as in the image segmentation examples, introduction of approximate iterations, such as the Fast Gauss Transform (reduces complexity to $O(N)$), use of suitable stopping criteria to eliminate unnecessary iterations, and employing other heuristic rules that save computations (checking conditions similar to the information force tree approach [28]). Also, while discussing the computational load, one should also consider the number of required iterations. From this aspect, using narrow kernels in mean shift and coupling this with a computationally inexpensive merging step is again preferable due to less number of iterations required as compared to using a wide kernel function in mean shift.

This paper lays the theoretical foundation for future work, where we will implement the techniques discussed above for further computational savings and an investigation of techniques for selecting a suitable threshold to identify mode connectivity for the purpose of automatic detection of the number of statistically significant clusters.

Acknowledgments: The authors would like to thank Miguel Carreira-Perpinan for valuable discussions. This work is partially supported by NSF ECS-0524835 and ECS-0622239.

REFERENCES

[1] M. Fiedler, "Algebraic Connectivity in Graphs," Czechoslovak Mathematics Journal, vol. 23, pp. 298-305, 1973.

[2] S. Sarkar, P. Soundararajan, "Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 52, pp. 504-525, 2000.

[3] J. Shi, J. Malik, "Normalized Cuts and Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, 2000.

[4] P. Perona, W. T. Freeman, "A Factorization Approach to Grouping," Proceedings of the European Conference on Computer Vision, pp. 655-670, 1998.

[5] M. Meila, L. Xu, "Multiway Cuts and Spectral Clustering," Technical Report 442, University of Washington, Department of Statistics, 2004.

[6] P. Chang, D. Schlag, J. Zien, "Spectral K-Way Ratio-Cut Partitioning and Clustering," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 9, pp. 1088-1096, 1994.

[7] R. Kannan, S. Vempala, A. Vetta, "On Clusterings: Good, Bad and Spectral," Proceedings of the IEEE Foundations of Computer Science, pp. 367-377, 2000.

[8] Y. Weiss, “Segmentation Using Eigenvectors: A Unifying View,” Proceedings of the International Conference on Computer Vision, pp. 975-982, 1999.

[9] C. Alpert, S. Yao, “Spectral Partitioning: The More Eigenvectors the Better,” Proceedings of the ACM/IEEE Design Automation Conference, pp.195-200, 1995.

[10] Y. Azar, A. Fiat, A. Karlin, F. McSherry, J. Saia, “Spectral Analysis of Data,” Proceedings of the Symposium on Theory of Computing, pp. 619-626, 2001.

[11] D. J. Higham, M. Kibble, “A Unified View of Spectral Clustering,” Technical Report 02, University of Strathclyde, Department of Mathematics, 2004.

[12] G. Scott, H. Longuet-Higgins, “Feature Grouping by Relocalisation of Eigenvectors of the Proximity Matrix,” Proc. of the British Machine Vision Conference, pp. 103-108, 1990.

[13] A.Y. Ng, M. Jordan, Y. Weiss, “On Spectral Clustering: Analysis and an Algorithm,” Advances in Neural Information Processing Systems, vol. 14, no. 2, pp. 849-856, 2001.

[14] R. Jenssen, T. Eltoft, J. C. Principe, “Information Theoretic Spectral Clustering,” Proceedings of the International Joint Conference on Neural Networks, pp. 111-116, 2004.

[15] K. Fukunaga, L.D. Hostetler, “The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition,” IEEE Transactions on Information Theory, vol. 21, pp. 3240, 1975.

[16] R. Jenssen, D. Erdogmus, J. C. Principe, T. Eltoft, “The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space,” Advances in Neural Information Processing Systems, pp.625-632, 2004.

[17] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790-799, 1995.

[18] D. Comaniciu, P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002.

[19] B. Georgescu, I. Shimshoni, P. Meer, “Mean Shift Based Clustering in High Dimensions: A Texture Classification Example,” Proceedings of the International Conference on Machine Vision, pp. 456-463, 2003.

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

[21] Carreira-Perpiñán, M. Á. and Williams, C. K. I. (2003): "On the number of modes of a Gaussian mixture". Scale-Space Methods in Computer Vision, pp. 625-640, Lecture Notes in Computer Science vol. 2695, Springer-Verlag.

[22] L. Devroye, G. Lugosi, *Combinatorial Methods in Density Estimation*, Springer, New York, 2001.

[23] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, MIT Press and Mc Graw-Hill, New York, 1990.

[24] E. Parzen, “On the Estimation of a Probability Density Function and the Mode,” Annals of Mathematical Statistics, vol. 32, pp. 1065-1076, 1962.

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

[26] L. Greengard and J. Strain, “The fast Gauss transform,” SIAM J. Sci. Statist. Comput., vol. 12, no. 1, pp. 79–94, 1991.

[27] M. Blatt, S. Wiseman, E. Domany, “Data clustering using a model granular magnet,” Neural Computation, vol. 9, no. 8, pp. 1805-1842, 1997.

[28] R. Jenssen, D. Erdogmus, K.E. Hild II, J.C. Principe, T. Eltoft, “Information Force Clustering Using Directed Trees,” Proceedings of CVPR’03, pp. 68-72, 2003.

[29] C. Yang, R. Duraiswami, L. Davis, “Efficient kernel machines using the improved fast gauss transform,” *Neural Information Processing Systems 17*, Cambridge, MA, 2005.

[30] J. Mercer, “Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations,” Transactions of the London Philosophical Society A, vol. 209, pp. 415-446, 1909.

[31] H. Weinert (ed.), *Reproducing Kernel Hilbert Spaces: Applications in Statistical Signal Processing*, Hutchinson Ross Pub. Co., Stroudsburg, Pennsylvania, 1982.

[32] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics,” International Conference on Computer Vision, 2001.

[33] D. Comaniciu, V. Ramesh, A. Del Bue, “Multivariate Saddle Point Detection for Statistical Clustering,” European Conference on Computer Vision (ECCV'02), Copenhagen, Denmark, vol. 3,pp. 561-576, 2002.

[34] Carreira-Perpiñán, M. Á., "Acceleration strategies for Gaussian mean-shift image segmentation". *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2006)*, pp. 1160-1167, 2006.

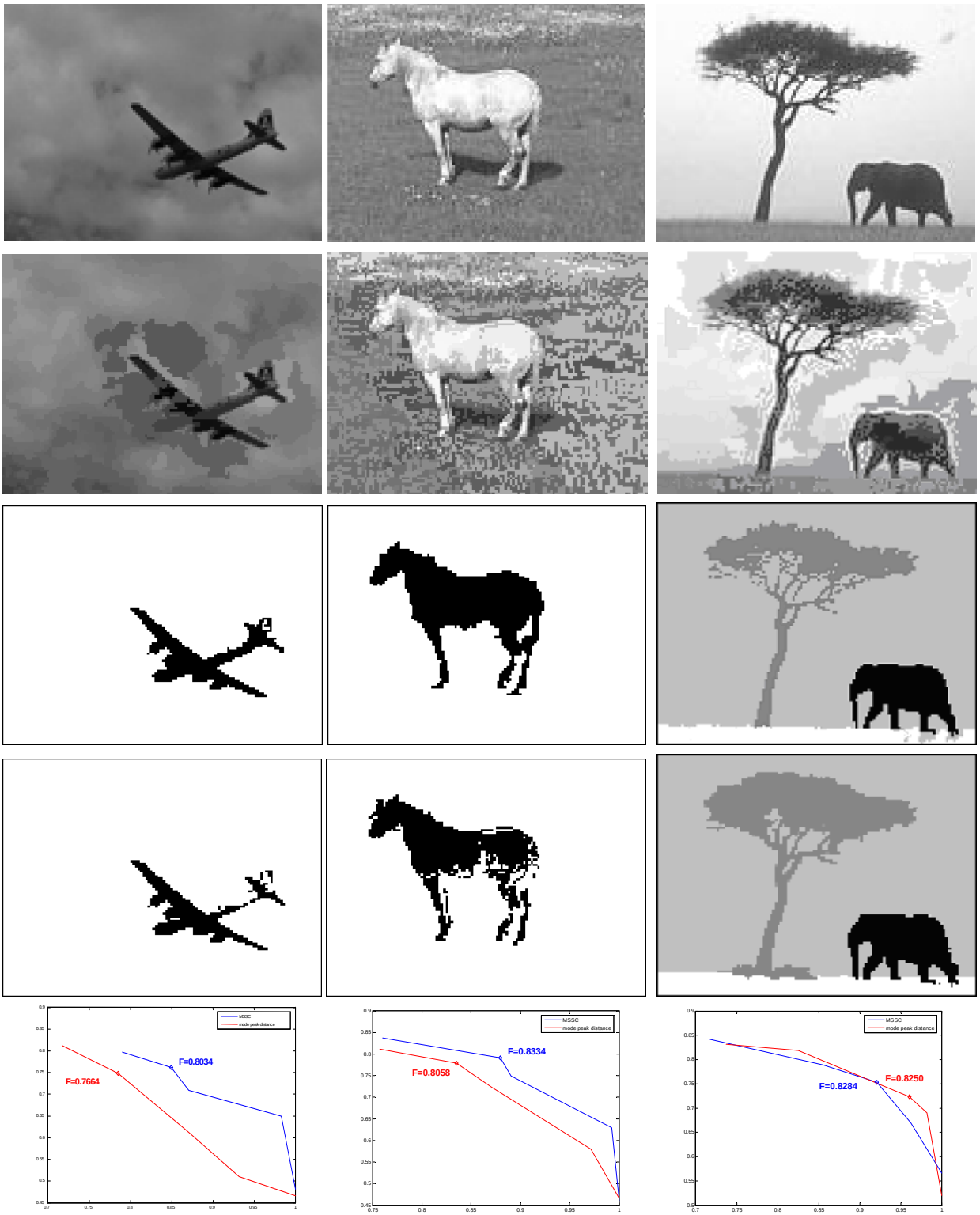

Figure 9. For each column the original image, the result of the mean shift step with the corresponding kernel size, the result of MSSC, the result of peak distances method, and corresponding precision-recall curves are presented, respectively. The kernel covariances used in the mean shift step for different points on the precision-recall curve is given by $\Sigma_i = \sigma^2 C_i$, where C_i is the local covariance estimate and $\sigma^2 = 0.25, 0.5, 1, 2, 4$.