Supervised and Unsupervised Clustering with Probabilistic Shift

Sanketh Shetty and Narendra Ahuja

Department of Electrical and Computer Engineering,
University of Illinois, Urbana-Champaign, Urbana, IL 61801 USA
{sshetty2,n-ahuja}@illinois.edu

Abstract. We present a novel scale adaptive, nonparametric approach to clustering point patterns. Clusters are detected by moving all points to their cluster cores using shift vectors. First, we propose a novel scale selection criterion based on local density isotropy which determines the neighborhoods over which the shift vectors are computed. We then construct a directed graph induced by these shift vectors. Clustering is obtained by simulating random walks on this digraph. We also examine the spectral properties of a similarity matrix obtained from the directed graph to obtain a K-way partitioning of the data. Additionally, we use the eigenvector alignment algorithm of [1] to automatically determine the number of clusters in the dataset. We also compare our approach with supervised[2] and completely unsupervised spectral clustering[1], normalized cuts[3], K-Means, and adaptive bandwidth meanshift[4] on MNIST digits, USPS digits and UCI machine learning data.

Keywords: Data Clustering, Image Segmentation.

1 Introduction

This paper is about automatic clustering with minimal user input. A cluster is viewed as a set of contiguous points having similar local point structures, defined by the point density, which are in contrast with their immediate surround. We allow clusters defined by a variety of global density-based criteria. (1) A cluster may consist of uniformly distributed points (having constant point density), or it may be characterized by a uniform density gradient, or it may be uniform in higher order derivatives of the density. (2) The gradient may be uniform along an open curve, giving rise to a uniform cluster. Alternately, an iso-density curve may be a closed contour in which case the cluster is modal, with a point of density extremum, surrounded by a succession of iso-contours with monotonically changing density. (3) A cluster may be of the same dimensionality as the underlying point pattern, or it may be confined to a subspace. (4) The defining criteria from (1) above, and other properties such as sizes, shapes and densities are unknown.

The basic idea of the proposed approach is to identify overlapping neighborhoods of points across the pattern, each completely contained within a cluster. Regardless of cluster type, we characterize these neighborhoods as density

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part V, LNCS 6315, pp. 644–657, 2010.

isotropic. Clearly, when a cluster has gradually varying density, the neighborhood size will be smaller - small enough to pass as isotropic within the tolerance level being used to test for density isotropy. Thus a cluster with arbitrarily complex, smoothly varying spatial density will be composed of overlapping neighborhoods each of whose size will be inversely proportional to the rate of local density change. Clustering then amounts to finding and distinctly labeling each connected set of overlapping, uniform-density neighborhoods. The connected components are extracted by letting each cluster implode to a dense core in its interior, thus resulting in as many well separated and uniquely labeled cores as the number of clusters. This is done by gradually moving each point within each cluster towards its core, by identifying a shift vector associated with the point which is directed towards the cluster core.

2 Related Work

Clustering algorithms are extremely diverse in their definition of clusters and approaches to finding them. Recent surveys of clustering algorithms are present in papers by Jain et al. [5] and Xu and Wunsch [6]. We restrict our discussion to algorithms relevant to motivating our approach, in particular the X-shift family of algorithms and spectral clustering algorithms.

The works of Fukunaga and Hostetler[7] and Koontz et al.[8] are early examples of clustering algorithms based on computing local density gradients. These techniques were rediscovered by the computer vision community in the recent past and applied to a host of problems in clustering image and video data. More recently methods based on computing a shift-vector based on mean[4], medoid[9] or median[10] of a point neighborhood have been proposed. The key idea is to compute a point or exemplar along the density gradient to which a point is shifted. Their advantages are that they are unrestricted in the shapes of the clusters and also automatically determine the number of clusters. The medoid-shift algorithm can also be applied to cases where only the distances or similarities between data is available. However, only adaptive bandwidth meanshift[11] addresses the problem of scale selection. Adaptive scales at individual points are computed using a pilot kernel density estimate obtained at a fixed scale K. We found the final clustering to be sensitive to this value of the initial bandwidth (see sec 5). Additionally, heuristics for merging modes and minimum cluster size significantly affect the final clustering. We differ from the X-shift algorithms in how our shift vector is computed. The X-shift algorithms move points along the density gradient towards the mode. However, they are not sensitive to other types of local density disparities that may exist in the data, e.g. a density step. This is because they rely on decisions that are local to a point neighborhood. In contrast, we rely on evidence accumulation from relevant adjacent neighbors to decide the local shift. X-shift methods are also likely to fail for clusters with uniform point distribution as a unique density mode is unlikely to exist. They return an oversegmented result for such clusters. In contrast, we model the isotropy of point distributions in local neighborhoods. We propose a statistical

testing approach to detect density isotropy. We are sensitive to relevant density changes e.g. cluster boundaries, density steps, and density gradients while ignoring incidental density disparities that may arise due to sampling, e.g. points in a uniform cluster. Subsequently, we use these detected density isotropic regions to determine a valid neighborhood over which each point influences its neighbors to shift in its direction. In section 5, we show qualitative and quantitative experiments that compare our shift vectors against those of X-shift algorithms.

Spectral approaches [12,2,3,1], involve the analysis of the graph Laplacian to obtain an embedding using its eigenvectors. Following this, regular K-means clustering or thresholding is applied to the embedded points to obtain a final clustering. Ng et al.[2] propose analyzing the symmetric, normalized graph Laplacian to obtain an embedding. The normalized cuts algorithm, in contrast can be viewed as analyzing eigenvectors of the transition probability matrix of a random walk on the undirected graph induced by the points[13]. Zelnik and Perona[1], address the problems of scale selection and automatic determination of the number of clusters for spectral clustering. The key advantage of these approaches lies in their ability to model clusters of unrestricted shapes in any subspace of the original space. However, Nader and Galun [14] construct several failure cases of such approaches, including the self-tuning spectral clustering algorithm. In particular, they identify problems with the scale selection parameter when there is a significant difference in density between adjacent clusters of different sizes. Additionally, these algorithms are sensitive to outliers in the dataset. We use the shift vectors computed using our approach to define a probabilistic directed graph. We analyze the spectral properties of affinity matrices derived from this digraph to obtain our final K-way and unsupervised clustering. This may be viewed as spectral clustering using an alternative graph construction technique. We demonstrate that this alternate construction, utilizing properties of shift vectors rather than K-nearest neighbors similarities, outperforms spectral clustering algorithms on real datasets.

3 Approach

Our proposed approach is an extension of the concept of the force transform, introduced in [15] for image analysis, to point sets in $\mathbb{R}^N$. The force transform produces a vector at each pixel, which represents the direction and magnitude of attraction experienced by the pixel from the rest of the image[15]. Region borders are identified as adjacent points with divergent vectors, whereas region skeletons are identified as adjacent points with convergent vectors. These vectors are computed at a set of spatial and image intensity scales, which are then used to produce a hierarchical image segmentation. Here our goal is to label points belonging to the cluster interior and border analogous to pixel labeling in image regions.

There are two major parts to our approach. The first part has to do with the detection of isotropic density neighborhoods. To this end, we use a test to determine if the neighborhood has isotropic point distribution in it. The

second part has to do with labeling connected components formed by overlapping isotropic density neighborhoods. This is made more complex than it may appear by the possibility of false detection or false rejection of an isotropic neighborhood, which may lead to cluster splits, e.g., in the neck area of a cluster, or cluster merges, e.g., in locally isotropic appearing neighborhoods between two distinct clusters. Although, postprocessing could be performed to detect and correct such errors, we have developed a formulation which avoids the need for such postprocessing by posing the problem as one of robust signal detection amidst noise in the first place. The signal here is the connected neighborhoods and the noise is deviations from density isotropy. We achieve this by iteratively, gradually and probabilistically shifting each point towards its cluster interior. This itself is done in two steps: by computing the local direction for shift, i.e., towards cluster interior, and then identifying the cluster (core) from these shift vectors.

Consequently, there are three major steps in our approach: (1) detection of density isotropic neighborhoods, (2) computation of shift vectors, and (3) identification of clusters utilizing probabilistic shift. The following subsections describe how we formulate each of these steps.

3.1 Detection of Isotropic Density Neighborhoods

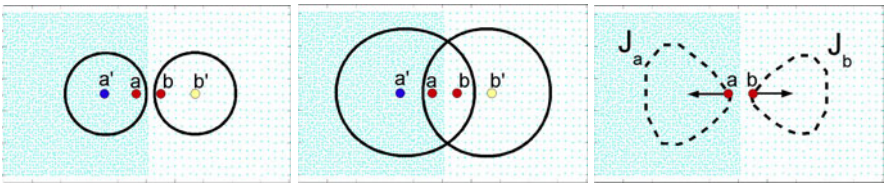
Our motivation for relying on isotropic density neighbors as the fundamental structures for clustering is as follows. It is reasonable to associate points within an isotropic density neighborhood with the same cluster. In contrast, density anisotropy, usually associated with a cluster boundary, indicates a plausible change in the cluster labels within a neighborhood. Therefore, density isotropy by itself may be used as a criterion for grouping points into clusters. However, we demonstrate that it is more useful as a scale selection criterion for computing shift vectors.

Force Criterion: We model the expected behavior of the force criterion [15] in isotropic and anisotropic neighborhoods to design a statistical testing approach to detect them. Figure 1(a) shows examples of isotropic density neighborhoods of a point. Figure 1(b) shows examples of anisotropic density neighborhoods of a point.

Given a set of points $\{\mathbf{x_i}\}_{i=1}^{n}$,centered at a point $\mathbf{y}$, and a weighting function $w(||\mathbf{y}-\mathbf{x}||)$, the force vector at $\mathbf{y}$ is computed as:

$$\mathbf{f}_n(\mathbf{y}) = \sum_{i=1}^{n} w(||\mathbf{y}-\mathbf{x_i}||) * \frac{(\mathbf{x_i}-\mathbf{y})}{||\mathbf{x_i}-\mathbf{y}||} \tag{1}$$

There are several possible choices for the weight function, $w(||.||)$. The only requirement is that it is non-increasing[15]. We denote the magnitude of the force over the n-nearest neighbors as $f_n(y) = ||\mathbf{f}_n(\mathbf{y})||$. Therefore, the set $\{f_i\}_{i=1}^{K}$ represents the magnitude of the force vector computed over increasing neighborhood sizes. We use this set to develop our criterion for detecting isotropic neighborhoods. A non-zero magnitude for the force vector indicates anisotropy



(a) (b) (c)

Fig. 1. Given two clusters with a density step between them, we show (a) isotropic density regions for points a' and b', to which points a and b belong. (b) However, the region for a' containing b has anisotropic density. Therefore, b does not belong to the influence neighborhood of a'. Similar reasoning holds for a and b'. (c) We show the sets J_a and J_b that contain a and b respectively in their influence neighborhoods. The shift is computed as a vector sum of influences of points in J.

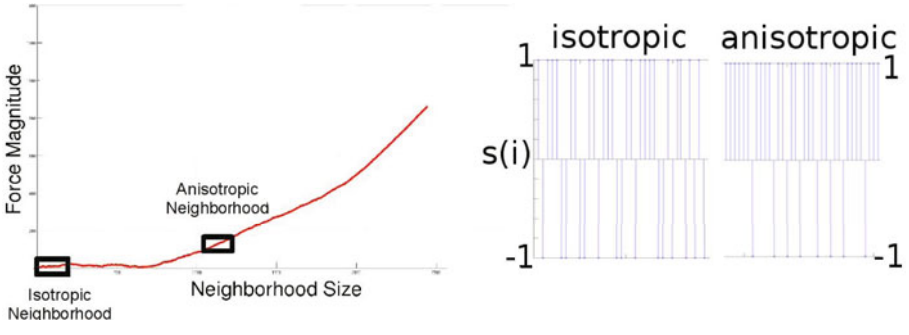
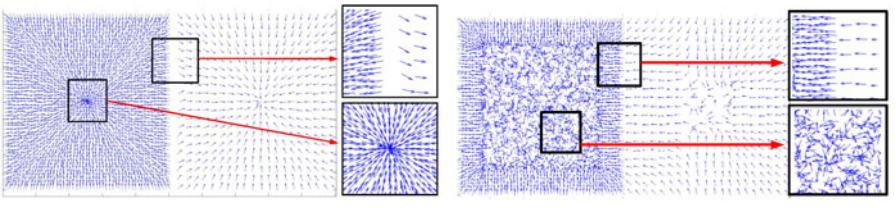


Fig. 2. (left) Plot of the force criterion from equation 1 over increasing neighborhood sizes. (right) Plot of random variable s_i for an isotropic density region and anisotropic density region.



(a) (b)

Fig. 3. (a) Shift vectors computed for the entire point set. Notice the shift vectors diverging at the cluster boundary and converging at the center.(b) Mean shift vectors for the same point set. They do not respect the density step between clusters and are arbitrarily oriented in the cluster interior. This results in cluster fragmentation.

in the distribution of points in the neighborhood. If the magnitude continues to increase as we grow the neighborhood around the point, it symptomatic of a growing anisotropy in the local point distribution. The force vector points in the direction of increasing point density. However, if the point distribution is symmetric we expect the magnitude to fluctuate. Figure 2 shows a plot of the force vector for different neighborhood sizes around a point of interest.

We define a random variable $s_i = sign(f_i - f_{i-1})$. This represents the sign of the difference of force magnitudes computed at two adjacent neighborhood sizes. We claim that in a region with isotropic point density the distribution of s_i is uniform at its two possible values $\{-1, 1\}$. In an isotropic region the force magnitude is as likely to increase as it is to decrease. Any anisotropy in the neighborhood is incidental unless it is statistically significant. Formally, we propose the identification of isotropic neighborhoods as a detection problem.

$$H_0 : \{s_i\}_{i=1}^{K} \; has \; zero \; median \tag{2}$$

$$H_1 : H_0 \; is \; false \tag{3}$$

We test for H_0 and H_1 using the sign test that this distribution has a zero median [16]. If H_0 is true it indicates an isotropic distribution of points in the given K-neighborhood. This test is performed at a significance level α. Therefore, for increasing neighborhood sizes we perform the sign test on the computed force magnitudes and return the first point of failure as the neighborhood size, K_i, over which the current point has influence. This is defined as the *influence neighborhood* of a point and is used to compute the shift vectors at points contained in it.

3.2 Shift Vector Computation

Let J_i denote the set of indices of points for which $\mathbf{x}_i$ appears within their respective neighborhoods of influence. It is reasonable for each point in J_i to assume that $\mathbf{x}_i$ shares its cluster label. However, it is also possible that for some $\mathbf{x}_i$, J_i has points from adjacent clusters, e.g., consider the case of points at the cluster boundary between two overlapping Gaussians. Therefore, we develop an approach where points in J_i compete for ownership of $\mathbf{x}_i$. The shift vector is the outcome of this competition. Given J_i the shift vector at a point is computed as:

$$\mathbf{a}_i = \sum_{j \epsilon J_i} w(||\mathbf{x}_i - \mathbf{x}_j||) * \frac{(\mathbf{x_j} - \mathbf{x_i})}{||\mathbf{x_j} - \mathbf{x_i}||} \tag{4}$$

Here $w(||.||)$ is some non-increasing weighting function. In our experiments we used the triangular weighting function ($w||.|| \propto 1\text{-}\frac{dist_j}{\max_{j \in J_i} dist_j}$, if $j \in J_i$, else 0). It is important to recognize the difference between the force vector $\mathbf{f}$, in section 3.1 , and the shift vector $\mathbf{a}$. The force vector, similar to X-shift vectors, points in the direction of the density gradient in the local neighborhood, as it is a purely local measure. In contrast, our shift vector, $\mathbf{a}$, points in the direction of greatest

agreement with local neighborhood properties. It points in the direction of the cluster whose points find the current point in most agreement with their local point distributions. This is important in our model of clustering as we seek to integrate neighborhoods with similar density properties while being sensitive to density discontinuities. Figures 3(a) and 3(b) further emphasize the advantages of our approach.

3.3 Cluster Identification

Cluster identification by connected components labeling of overlapping uniform neighborhoods has been proposed in [17]. However, as stated at the beginning of this section, this may lead to cluster splits and merges . Our shift vectors allow for a more informed connected components labeling. Shift vector at a point is directed in the general direction of the cluster core. We propagate labels in the general direction of the shift vector. We construct a probabilistic directed graph by connecting each point to other points in its influence neighborhood that lie in the half-space in the direction of its shift vector. Points are shifted probabilistically along this graph to cluster cores where the final clusters are obtained. This is realized using the interpoint transition probability matrix for the points defining the digraph.

Constructing the Transition Probability Matrix. First, define a directed graph G = <X,P>, composed of a node set X = $\{\mathbf{x}_i\}_{i=1}^{n}$ and a transition probability matrix P = $\{p_{ij}$: probability of a transition from node i to node j$\}$. Given a node $\mathbf{x}_i$, its shift vector $\mathbf{a}_i$, and its influence neighborhood set K_i, we define a variable $t_{ij} \in [0, 1]$ which represents the preference for moving from node i to node j. Formally, it is defined as:

$$t_{ij} = \max(0, w(||\mathbf{x}_j - \mathbf{x}_i||)\langle \mathbf{a}_i, \mathbf{x}_j - \mathbf{x}_i \rangle) \tag{5}$$

This produces positive values for nodes in the positive half space of the hyperplane defined by $\mathbf{a_i}$ at $\mathbf{x_i}$ and 0 otherwise. From this we obtain the probability of transitioning to a node $\mathbf{x}_j, j \in K_i$:

$$p_{ij} = \frac{t_{ij}}{\sum_{j \epsilon K_i} t_{ij}} \tag{6}$$

Consider a point $\mathbf{x}_b$ near the cluster border. Its shift vector, by construction, points towards the cluster interior. Therefore, for a point $\mathbf{x}_j$, within its influence neighborhood in the cluster interior, there is a non-zero probability of a transition from $\mathbf{x}_b$ to $\mathbf{x}_j$. However, the reverse is not necessarily true as $\mathbf{x}_b$ is unlikely to lie in the positive half-space of $\mathbf{a}_j$. In contrast, shift vectors for points in the cluster core, converge. Each core member lies in the positive half-space of shift vectors of several other core points. This generates a non-zero probability of transition between nodes in the core, making its constituents nodes in a strongly connected subgraph of G. Since, P represents the transition probability matrix

for this digraph G, taking powers of P simulates random walks on the graph G. It is straightforward to see that these random walks move points from the cluster boundary towards the core of the cluster. The preference for points to transition to the cluster boundary disappears after a few iterations.

The row $\mathbf{p}_i^N$ of the P^N, represents the probability of a walk starting at $\mathbf{x}_i$ transitioning to other nodes in the graph over N steps. Once the walk transitions to the core to of the cluster, this transition probability vector begins to converge to a steady state value. This is a direct consequence of the core of the cluster being a strongly connected subgraph. We denote the transition probability matrix, with rows that have converged to a steady state value, as P^ϵ. We obtain this matrix by multiplying out the rows $\mathbf{p}_i$ repeated with P until, the normed difference in probability distributions between consecutive iterations is less than ϵ. We denote this final probability vector as $\mathbf{p}_i^\epsilon$. There are two interpretations of the entries in $\mathbf{p}_i^\epsilon$: (1) we view, $j = \arg\max(\mathbf{p}_i^\epsilon)$ as the most likely final destination of a walk starting at $\mathbf{x}_i$ and use this to perform a connected components labeling. (2) Alternatively, $\mathbf{p}_i^\epsilon$ may be viewed as a soft assignment of final destinations of a walk originating at $\mathbf{x}_i$. We can compare distributions of $\mathbf{p}^\epsilon$ for different nodes to construct an affinity matrix. We perform spectral clustering on this matrix to give us the final clustering.

4 Algorithms

4.1 Partitioning by Connected Destinations

Given P^ϵ, a straightforward algorithm is to assign each node to its most probable destination. Nodes with the same destination are grouped in the same cluster. However, it is possible that some of the destinations themselves converge on other nodes. Therefore, a simple connected components labeling algorithm is executed to obtain the final labeling. We will refer to this algorithm as Clustering with Shift Vectors (CSV).

4.2 Supervised and Unsupervised Spectral Partitioning

Assuming points with similar probability distributions of their final destinations are more likely to belong to the same cluster, we can obtain a similarity matrix for our data by comparing these distributions. Clustering is obtained by spectral analysis of this similarity matrix. Alternatively, the similarity matrix can be constructed based on the initial transition probability distributions P. We discuss both alternatives here.

Given P, each row represents a probability mass function for a corresponding node in G transitioning to other nodes in its influence neighborhood. Let $\hat{P}$ denote the row-normalized version of the matrix P. We can define an similarity matrix based on $\hat{P}$ as: $A^I = \hat{P}\hat{P}^T$.

Nodes with preferences to transition to similar parts of the cluster interior have a higher similarity than nodes which transition to other clusters or other parts of the same cluster interior. Consequently, the matrix A^I is very sparse.

Similarly, given P^ϵ, each row represents a probability mass function (PMF) for the final destination of the corresponding point. Points with similar PMF's are more likely to belong to the same cluster (or same part of the cluster) than points with different PMF's. We use this intuition to define a similarity matrix A^ϵ as: $A^\epsilon = \hat{P}^\epsilon \hat{P}^{\epsilon T}$.

We observed that A^ϵ is blockwise dense. Nodes in a cluster usually converge to the same subset of nodes in the core. Therefore, the similarities of their PMF's are likely to be very high.

In the **supervised** setting, the user specifies the number of partitions, K. We perform a K-way graph partitioning following the method of [2]: (1) Compute the normalized laplacian $L = D^{-\frac{1}{2}} A^I D^{-\frac{1}{2}}$(or A^ϵ). Here D denotes the degree matrix.(2) Compute the top K eigenvectors of L and stack them columnwise in a matrix E. (3) Normalize rows of E. (4) Perform K-means clustering on rows of E to obtain final clustering. We refer to this algorithm as the Spectral Clustering with Shift Vectors (SCSV-K).

In the **unsupervised** setting we adopt the eigenvector alignment algorithm proposed in [1] to automatically determine the number of clusters: (1) Given choices for number of clusters $K_c = \{K_1 ... K_m\}$ compute the top $\max(K_c)$ eigenvectors of the normalized laplacian of A^I (or A^ϵ). (2) For each column subset $1 : K_c(i)$ of the eigenvector matrix E, compute the rotation that best aligns this column with the canonical coordinates, by gradient descent. (3) Score the alignment based on the distortion measure [1], to obtain $C_{K_c(i)}$. (4) Return the number of clusters as the number of columns with the best alignment score. (5) Stack corresponding columns to form E and normalize its rows. (6) Return the final clustering as the output of K_{best}-means algorithm. We refer to this algorithm as Zelnik-Perona Clustering with Shift Vectors (ZPCSV). We preprocess both P and P^ϵ to remove outliers by removing all nodes with zero transition probabilities to other nodes.

5 Results

In this section we present the results of experiments with three variants of our algorithm: Clustering with Shift Vectors (CSV), Spectral Clustering with Shift Vectors (SCSV-K) and Zelnik-Perona Clustering with Shift Vectors (ZPCSV). We first present qualitative results on challenging artificial datasets. We then compare the SCSV-K algorithm against K-Means(KM), Locally Scaled Spectral Clustering (ls-SC)[1] and Normalized Cuts (NC)[2]. We compare our unsupervised algorithms, CSV and ZPCSV, against Adaptive bandwidth Meanshift (AMS)[3] and the Zelnik-Perona Spectral Clustering (ZPC)[1].

Implementation Details: We do not address the issue of selection of an optimal α parameter for testing density isotropy, for a dataset. We expect it to

[1] `http://webee.technion.ac.il/~lihi/Demos/SelfTuningClustering.html`
[2] `http://www.cis.upenn.edu/~jshi/software/`
[3] `http://www.caip.rutgers.edu/riul/research/code/AMS/index.html`

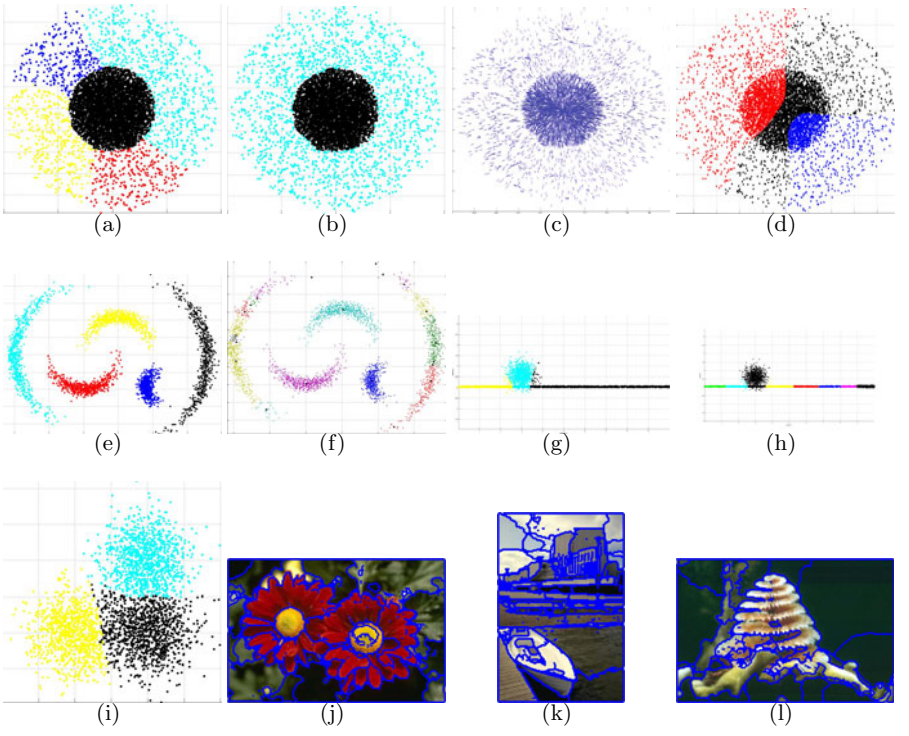


Fig. 4. (a) Output of our CSV algorithm on overlapping clusters with different densities. (b) Output of the ZPCSV algorithm which determines the right number of clusters based on the criterion described in [1]. (c) Shift vectors computed by our method. (d) Output of Adaptive bandwidth Mean shift on the same data. (e) Our performance on the crescent dataset from [9] and (f) the output of medoid-shift for an arbitrary bandwidth setting. (g) Clusters detected by ZPCSV for a Gaussian overlapping with an elongated uniform cluster from [14]. (h) The output of Zelnik-Perona Clustering on the same data. (i) Output of CSV on three overlapping Gaussian clusters data.(j-l) The output of CSV for color image segmentation. Notice that outliers are detected as isolated pixels within image segments.

be a function of degree of sampling in the data, but this discussion is beyond the scope of this paper. To deal with the diverse datasets in our experiments we computed shift vectors at $\alpha = \{0.05, 0.025, 0.01, 0.0075, 0.005, 0.0025, 0.001\}$, for each point. The final shift vector at a point is computed as the vector sum of shift vectors obtained at that point, at each significance level. We used the uniform weighting function $w||.|| = 1$ for computing forces (equation 1), and the entries in the transition probability matrix (equation 5). We used the triangular weighting kernel in equation 4. We specified the number of clusters between 1 and 20, for the unsupervised spectral clustering algorithm to evaluate its cost function. We set $\epsilon = 5e - 4$. These settings were used for all experiments with artificial and real data.

5.1 Artificial Data

The experiments on artificial datasets demonstrate the ability of our approach to cluster (1) both uniform and modal clusters(fig.4(a)-4(b),4(i)), (2) multiscale clusters(fig.4(b)), (3) overlapping clusters of different densities (fig. 4(a)-4(b)) and (4) clusters with arbitrary shapes (fig. 4(e)). (1), (2) and (3) are challenging cases for X-shift based algorithms as shown in figure 4(d). Figure 4(b) shows that ZPCSV, is a reasonable approach to utilizing outputs of our probabilistic shift algorithm to identify the correct number of clusters in data. The corresponding output of CSV is shown in figure 4(a). Figure 4(f) shows that though X-shift based approaches can detect arbitrarily shaped clusters, this too relies on a proper bandwidth setting. In contrast, we use the same parameter settings across all datasets, demonstrating our invariance across a wide variety of data. We compare the outputs of ZPC and ZPCSV on a dataset with a gaussian overlapping an elongated uniform cluster, a challenging dataset from [14]. Self-tuning clustering using the scaled K-NN kernel oversegments the dataset (fig 4(h)). However, the ZPCSV algorithm accurately picks the right number of clusters, while accounting for the density discontinuity that arises when the two distributions overlap (fig 4(g)). This demonstrates that the affinity matrix obtained through probablistic shift captures local geometry better than the locally-scaled affinity matrix suggested in [1].

5.2 Real Data

We use real data to provide quantitative comparisons between our approach and popular algorithms in literature. We use USPS digits, MNIST digits, and datasets from the UCI Machine Learning repository for comparing algorithms. We also demonstrate an application of our algorithm to segment color images (figures 4(j)-4(l)).

Measuring Clustering Accuracy: For all our evaluation tasks, we work with data for which the labeling is known. We define clustering accuracy as follows. For each cluster detected, we check the number of unique "ground-truth" labels present. Next we determine the label class with maximum representation within each cluster. The remaining points in the cluster are identified as being wrongly clustered. The clustering error is the percentage of points in the dataset that are assigned to wrong clusters.

Digits Data: We use 9268 USPS digits (16×16 images digits 0-9) and 10000 MNIST digits(28×28 images of 0-9) to compare the performance of clustering algorithms. These datasets pose an interesting challenge. The same digit written by different people is likely to be more similar to other digits from the same class, producing distinct clusters. However, some digits have very similar appearances, e.g. 4's and 9's, and produce overlapping clusters.

We vectorize the digit images from USPS into 256 and MNIST into 784 dimensions. In the first set of experiments, within each dataset, we gave each of the 45 possible pairs of digits as inputs to the clustering algorithms(e.g. 0's vs. 1's).

We report average performance of each algorithm on these 45 pairs in tables 1 and 2. SCSV-K outperforms all other supervised clustering algorithms on both datasets (table 1). This indicates the affinity matrix constructed using shift vectors produces a more accurate picture of cluster structure. Similarly, ZPCSV outperforms all other unsupervised clustering algorithms. ZPCSV is not restricted to finding 2 clusters in the experiments, therefore it can find multiple clusters within a single digit. However, the median number of clusters returned was 2 for both datasets. CSV shows similar performance however the median number of clusters was 3 for MNIST and 4 for USPS, indicating a tendency to fragment clusters. On average we found 7 outliers (out of 1500-2000 points) per experiment for both datasets. We also experimented with giving all 10 classes simultaneously to the clustering algorithms. Here too SCSV-K outperformed other supervised clustering algorithms. Among unsupervised algorithms CSV performed the best (table 2). Comparatively, ZPCSV performs worse because it finds fewer clusters than there are classes in the data. However, it still beats the original ZPC on both datasets, reinforcing our claim that we construct better affinity matrices using shift vectors. To compare our results against adaptive bandwidth meanshift we varied the initial bandwidth at samples between 10 and 1500 nearest neighbors. We report the best results obtained for their algorithm over this range. We used this best performing bandwidth setting for experiments with all digits. AMS performed worse than all other algorithms compared. Curiously, when we performed K-means clustering on the modes to which individual points in AMS converge, we obtained better results. For example, in the USPS digits clustering task with 10 classes, we obtained an accuracy of 26.52% with K=10, when we post-processed the converged AMS modes using K-means. We attribute this variation in performance to the heuristics employed in merging modes and specifying a minimum cluster size. In contrast we do not invoke heuristics to post process our clustering.

UCI ML: We also tested our algorithms on data sets from the UCI Machine Learning Repository(see tables). Our K-way algorithm outperformed other clustering algorithms on most tasks. Interestingly, we noticed that our performance on the SVMGUIDE dataset improved significantly we performed spectral analysis on our the affinity matrix obtained from the initial transition matrix P, instead of using P^ϵ. These gains were not significant for other tasks. Although CSV returned 2 clusters, one of them contained over 95% of the points in the dataset. We obtained lower error rates with stricter α criterion. These results suggest that direct analysis of P avoids the rare cases where CSV fails to find the right cluster cores. This is because the affinity matrix computed using P relies on the local correlations of shift vectors, as opposed to the final destinations of the shift. It should be noted that spectral clustering using the affinity matrix obtained from P, consistently outperforms all other spectral clustering techniques(table 2). From the table it appears that on an unknown dataset CSV would give lower expected error than other methods discussed here.

Table 1. Comparison of supervised clustering errors (%) for various datasets. The average error is reported for pairwise experiments with MNIST digits and USPS digits.

Data	KM	NC	ls-SC	SCSV(K)	
				P^ϵ	P
MNIST(Pairs)	9.04	8.8	8.3	**2.7**	2.77
MNIST(All)	38.54	80.56	59.49	**16.2**	17.3
USPS(Pairs)	7.54	5.1	5.13	**0.89**	0.976
USPS(All)	26.51	30.07	50.1	**10.92**	18.82
Ionosphere	28.8	10.83	10.82	**3.65**	**3.65**
Breast-Cancer	3.95	34.99	34.99	4	**3.5**
Diabetes	34.9	34.9	**34.51**	34.85	34.85
SVM-Guide	23.5	12.42	19.61	43.42	**5.85**

Table 2. Comparison of unsupervised clustering errors (%). The numbers in brackets are the number of clusters detected.

Data	AMS	ZPC	ZPCSV		CSV
			P^ϵ	P	
MNIST(Pairs)	45.2	7.8	**1.42**	1.62	2.48
MNIST(All)	79.3(9)	80.2(2)	49.87(5)	40.23(7)	**17.2**(12)
USPS(Pairs)	38.8	3.84	**0.913**	0.987	0.984
USPS(All)	71.9(9)	69.9(2)	18.75(8)	25.61(7)	**4.7**(15)
Ionosphere	10.86(3)	10.83(15)	**3.65**(9)	**3.65**(10)	**3.65**(2)
Breast-Cancer	26.35(5)	**3.2**(7)	4(2)	3.5(2)	3.83(8)
Diabetes	34.37(3)	33.98(2)	**33.55**(10)	34.85(2)	33.81(4)
SVM-Guide	**4.8**(12)	9.03(8)	43.42(10)	6.46(6)	43.22(2)

6 Conclusions and Contributions

This paper makes three chief contributions. (1) We have introduced a novel scale selection criterion based on density isotropy for the computation of shift vectors. (2) Probabilistic shift using these shift vectors are shown to perform better than X-shift methods on both real and challenging artificial datasets. (3) Affinity matrices computed using these shift vectors are shown to consistently outperform both supervised and unsupervised spectral clustering algorithms. We argue this is a direct consequence of the principled evidence accumulation approach adopted to determine local shift properties for points. One drawback of the probabilistic shift approach is the computational complexity of computing P^ϵ ($\mathrm{O}(N^3)$) by matrix multiplication. In future work we will explore avenues to compute this efficiently or to approximate it.

Acknowledgement. The support of the Office of Naval Research under grant N00014-09-1-0017 and the National Science Foundation under grant IIS 08-12188 is gratefully acknowledged.

References

1. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems, vol. 17, pp. 1601–1608 (2004)
2. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in NIPS, vol. 14 (2002)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
4. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
5. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. Surveys 31(3), 264–323 (1999)
6. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16 (2005)
7. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 21, 32–40 (1975)
8. Koontz, W.L.G., Narendra, P.M., Fukunaga, K.: A graph-theoretic approach to nonparametric cluster analysis. IEEE Trans. Comput. 25, 936–944 (1976)
9. Sheikh, Y.A., Khan, E., Kanade, T.: Mode-seeking by medoidshifts. In: Eleventh IEEE International Conference on Computer Vision (2007)
10. Shapira, L., Avidan, S., Shamir, A.: Mode-detection via median-shift. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1909–1916 (2009)
11. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 438–445 (2001)
12. Chung, F.R.K.: Spectral Graph Theory. CBMS Regional Conference Series in Mathematics, vol. 92. American Mathematical Society, Providence (1997)
13. Maila, M., Shi, J.: A random walks view of spectral segmentation. In: AI and STATISTICS (AISTATS) (2001)
14. Nadler, B., Galun, M.: Fundamental limitations of spectral clustering. In: Advances in NIPS, pp. 1017–1024 (2007)
15. Ahuja, N.: A transform for multiscale image segmentation by integrated edge and region detection. PAMI 18, 1211–1235 (1996)
16. Gibbons, J.D.: Nonparametric Statistical Inference. Marcel Dekker, New York (1985)
17. Shetty, S., Ahuja, N.: A uniformity criterion and algorithm for data clustering. In: Proceedings of the 19th ICPR (2008)