Robust clustering using a kNN mode seeking ensemble[1]

Jonas Nordhaug Myhre[a,c,*], Karl Øyvind Mikalsen[a,c], Sigurd Løkse[a,b], Robert Jenssen[a,b]

[a]*Machine Learning @ UiT Lab - site.uit.no/ml*
[b]*Department of Physics and Technology, UiT - The Arctic University of Norway, Tromsø, Norway*
[c]*Department of Mathematics and Statistics, UiT - The Arctic University of Norway, Tromsø, Norway*

Abstract

In this paper we present a new algorithm for parameter-free clustering by mode seeking. Mode seeking, especially in the form of the mean shift algorithm, is a widely used strategy for clustering data, but at the same time prone to poor performance if the parameters are not chosen correctly. We propose to form a *clustering ensemble* consisting of repeated and bootstrapped runs of the recent kNN mode seeking algorithm, an algorithm which is faster than ordinary mean shift and more suited for high dimensional data. This creates a robust mode seeking clustering algorithm with respect to the choice of hyper-parameters and high dimensional input spaces, while at the same inheriting all other strengths of mode seeking in general. We demonstrate promising results on a number of synthetic and real data sets.

Keywords: Density based clustering, Consensus clustering, kNN mode seeking, Mean shift, Ensemble clustering,

1. Introduction

Density based clustering is one of the fundamental directions in unsupervised learning [2, 3, 4, 5, 6, 7, 8, 9]. The natural notion of a *cluster* consisting of points gathered together in regions of high probability is very intuitive, and forms the basis of density based clustering. Furthermore, one of the prominent methods of density based clustering is *mode seeking*, most often represented by the *mean shift* algorithm [10]. In mean shift and mode seeking in general each data point is connected to a mode (local maximum) of the probability density function (pdf) and each mode represents a cluster. Successful applications of mean shift include Microsoft's Kinect® computer vision system [11], object motion tracking [12], initalization of spectral clustering algorithms [13, 14] and change detection in satellite radar images [15]. In addition more recent applications have shown good results in visualizing functional connectivity in the brain [16], semi-supervised learning [17] and fault detection [18]. Mode

[1]This work is an extended version of [1], presented at the Scandinavian Conference on Image Analysis 2015.

*Corresponding author

Email addresses: jonas.n.myhre@uit.no (Jonas Nordhaug Myhre), karl.o.mikalsen@uit.no (Karl Øyvind Mikalsen), sigurd.lokse@uit.no (Sigurd Løkse), robert.jenssen@uit.no (Robert Jenssen)

seeking for clustering based on mean shift is therefore a very influential methodology that has been very useful for solving real world problems. There are however a variety of problematic issues, that if solved, would make mode seeking based clustering even more powerful. Mean shift is for example very sensitive to user defined parameters that greatly influence the number of clusters returned by the method. Moreover, mean shift is slow, and does not scale well as the number of dimensions (features) increases. Some approaches for mitigating such effects have been proposed in recent years [19, 20, 21, 22, 23, 24, 25, 26, 27, 28], but the problematic issues still remain to a large degree.

In this paper, our aim is to take mode based clustering further by proposing a different strategy. In our approach, traditional mean shift is replaced altogether by a much faster k nearest neighbor (kNN) mode seeking method [29]. In addition, ideas from ensemble (consensus) clustering are incorporated for robustness with respect to hyperparameters. These choices are motivated and explained below.

kNN mode seeking [29] is a recently proposed alternative method for mode seeking clustering which is significantly faster than mean shift. At the same time it retains comparable accuracy to ordinary mean shift [29]. The algorithm is based on the same principles as mean shift, namely following the local gradient ascent of each point and using the mode it converges to as cluster indicator. The difference lies in the underlying density estimate. While mean shift uses a kernel density estimate, kNN mode seeking uses a kNN density estimate [30], a more adaptive but less smooth estimate. Additionally, the modes and the gradient ascent path connected to the modes are relaxed to only consist of points available in the input data set[2]. The latter property gives an advantage in terms of computational complexity that becomes even clearer in high-dimensional space.

Unfortunately, the kNN mode seeking algorithm is not free of the main problem of probability density estimation; selecting the critical bandwidth parameter of the density estimate. In the case of kNN mode seeking it is the neighborhood size parameter k. A poorly chosen nearest neighborhood parameter (kernel density bandwidth for mean shift) leads to an underlying probability density estimate that does not represent the data well. Choosing a too small neighborhood gives a spurious and spiky density with too many local modes. On the other hand, a too large neighborhood size will oversmooth the density leading to a single all-encompassing cluster in the limit. This problem gets exponentially harder as the dimensionality increases due to the fact that in most cases with a bounded number of data points, high dimensional data spaces are mostly empty [3, 31].

In this work, inspired by the ideas of *consensus clustering* [32, 33, 34, 35], we propose to execute the kNN mode seeking several times using a varying neighborhood parameter and let the combined results form an *ensemble* that returns the final clustering. The key idea of consensus clustering, also known as ensemble clustering, is that multiple runs of the same algorithm with different initializations or different parameters will create a more diverse representation of the underlying structure of the data [33, 2]. This will in turn give a more stable and robust final clustering result.

A problem that arises when incorporating mode seeking clustering into consensus clus-

[2]See Figure 2 on page 6 for an illustration.

tering is that the modal clustering algorithms mentioned here, kNN mode seeking and mean shift, are both deterministic. A key component in consensus clustering is diversity achieved by the inherent randomness in the clustering algorithms. Thus, multiple runs without parameter change will yield the same result. As a solution, we propose to perform random subsampling without replacement for each repeated clustering. This technique, which is similar to bootstrap aggregation used in random forests [36, 37] is well established and has previously showed promising results for clustering gene expression data using a collection of self-organizing maps and average linkage hierarchical clustering [33].

As we will show, our contributions will increase the robustness of the kNN mode seeking algorithm towards local variation of different scales. In doing this we take a big step towards a more user-friendly clustering scheme where *manual parameter tuning is not necessary.*

To summarize, our contributions are: (1) A new robust algorithm for parameter free kNN mode seeking clustering capable of fast high dimensional clustering. (2) Introducing ensemble clustering to improve mode based clustering. (3) Introducing several procedures for introducing randomness into the mode seeking ensemble framework[3].

The remainder of this paper is organized as follows. In Section 2 and 3 we review clustering by mode seeking and the kNN mode seeking algorithm. Section 4 introduces our proposed clustering algorithm. Finally, Section 5 shows experimental results on both real and synthetic data.

2. Background: Clustering by mode seeking.

Mean shift and kNN mode seeking both fall in the category *mode based clustering*, or simply *modal clustering* [2, 3]. The essence of mode based clustering is gradient ascent on the underlying pdf of the data one wants to analyze. Originally introduced by Cheng, [38], it was revitalized and made popular through the *mean shift* algorithm [10]. The main idea is as follows; given a pdf, most often through an estimate, each point in the support of the pdf will have an integral curve that converges to a mode of the pdf, see for example [39]. All points that converge to the same mode are considered part of the same *cluster*. This induces a partition over the support of the density. The part of the support that converges to the same mode is known as the *basin of attraction* of the mode.

The benefits of using modal clusterings are many, the density can adapt locally to the data allowing to some extent to capture nonlinear clusters, the number of clusters is determined by the density estimate. It is also robust to outliers, an outlier will represent its own cluster and can be thresholded away based on its density value.

A mode can be defined to be a point at which the pdf has a local maximum. However, to avoid confusion in cases where the mode can be represented by several points (most notably a uniform density) we include a more general definition

Definition 1. *Let $f : X \in \mathbb{R}^d \to \mathbb{R}$ be a pdf. A connected set of points $\mathcal{M} \subseteq X$ is a mode of f if $\forall m \in \mathcal{M}$, $f(m) = h > 0$ and there exists a compact, connected set $C_{\mathcal{M}} \supset \mathcal{M}$ with $\partial C_{\mathcal{M}} \cap \mathcal{M} = \emptyset$, where $\partial C_{\mathcal{M}}$ is the boundary of $C_{\mathcal{M}}$, such that $f(x) \leq h$, $\forall x \in \mathcal{C}_{\mathcal{M}}$.*

[3]This paper is an extension of previous work [1]

This definition allows a mode to consist of several connected points, such that uniform areas of a density, for example a plateau or a flat ridge of equal probability density, is covered by the definition.

More concretely, in the mean shift algorithm [10, 38], the pdf of a data set is estimated by a *kernel density estimate* (KDE), [30], and gradient ascent is performed on the estimated pdf.

The standard kernel density estimator for a set $X = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d$ is given as follows:

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i), \tag{1}$$

where $k(\cdot, \cdot)$ is a symmetric positive (semi-)definite function integrating to one. The most commonly used kernel function is the Gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{c_g} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right). \tag{2}$$

Here σ^2 is the bandwidth of the kernel, $\|\cdot\|$ denotes Euclidean norm ad c_g is a normalizing constant ensuring that the kernel density integrates to one.

The gradient of the KDE is given as :

$$\nabla \hat{f}(\mathbf{x}) = -\frac{1}{n} \sum_{i=1}^{n} \frac{(\mathbf{x} - \mathbf{x}_i)}{\sigma^2} k(\mathbf{x}, \mathbf{x}_i), \tag{3}$$

which is proportional to the *mean shift* vector:

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i k(\mathbf{x}, \mathbf{x}_i)}{\sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i)} - \mathbf{x}. \tag{4}$$

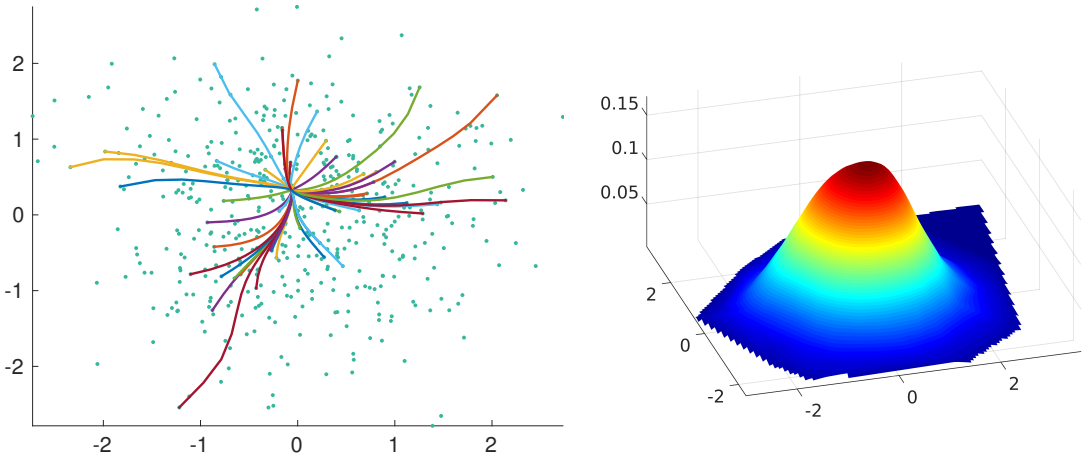
See Comaniciu et al. [10] for further details. Note that normalizing constants can be omitted, as gradient ascent is only dependent on the *direction* of the gradient.

To summarize, the mean shift clustering algorithm can be stated as follows:

1. For each input point $\mathbf{x}_i$ and a threshold ε:
   - While $\|\mathbf{m}(\mathbf{x}_i)\| > \varepsilon$, take a step in the direction of the mean shift vector, $\mathbf{x}_i \leftarrow \mathbf{m}(\mathbf{x}_i)$.
2. Assign each point $\mathbf{x}_i$ that has converged to the same mode to the same cluster C

In Figure 1 we see an example showing the mean shift procedure on a random selection of 50 points from a sample consisting of 500 points from a bivariate standard normal distribution. We see that the trajectories (left) are smooth paths from each input point converging to the mode of the kernel density estimate which is shown in Figure 1b.

The computational complexity of mean shift is $\mathcal{O}(Tn^2)$, where n is the number of samples and T is the number of initial points (typically chosen as $T = n$). We should also note that each mean shift trajectory, T in total, is independent of all others so the algorithm is trivially



(a) Mean shift trajectories.

(b) Kernel density estimate.

Figure 1: (a)Mean shift trajectories on a random normal sample. The kernel bandwidth σ^2 was chosen equal to the average distance to the 20th nearest neighbor. The colors of the trajectories are chosen at random so that they are easier to separate visually. We note that all trajectories converge to the same mode in this example. (b) 3D plot of the estimated density.

parallelizable allowing for considerable speedup when multiple threads are available. Also, if the dimensionality of the data is extremely high, the running time of calculating Euclidean distances used in the kernel density estimate have to be considered.

Furthermore, it is obvious that a good choice of bandwidth parameter σ^2 is essential to obtain a good clustering result. If we consider the example of a sample from a standard normal distribution a too small bandwidth will result in many small clusters – equal to the number of samples in the most extreme case. On the other hand, setting the bandwidth too large will result in oversmoothing of the density. In the unimodal case oversmoothing is not necessarily a problem, since even though the density may represent nonlinear structure with a single global maximum an oversmoothed density would represent the correct clustering, perhaps with some bias [40, 41].

Finally, a closely related problem is the ability of the underlying density to capture the structure of the data in smooth high density regions. Recall that for the mean shift to cluster a collection of data points together, they need integral curve trajectories that converge to the same local maximum. For this to work, the kernel density estimate have to capture unimodal dense regions that corresponds to the data clusters, whichis not trivial when the clusters have shapes that are nonlinear or consists of overlapping mixture distributions [42].

3. kNN mode seeking

The kNN mode seeking algorithm is a mode based clustering algorithm where the kernel density estimate is replaced by a kNN density estimate [29, 30]. It was introduced by Koontz et al. [43] and reformulated by Duin et al. [29]. In addition to replacing the kernel density with a kNN density estimate, the gradient ascent integral curve approximations are reduced to consist only of data points contained in the input data set. As a result of this, the modes

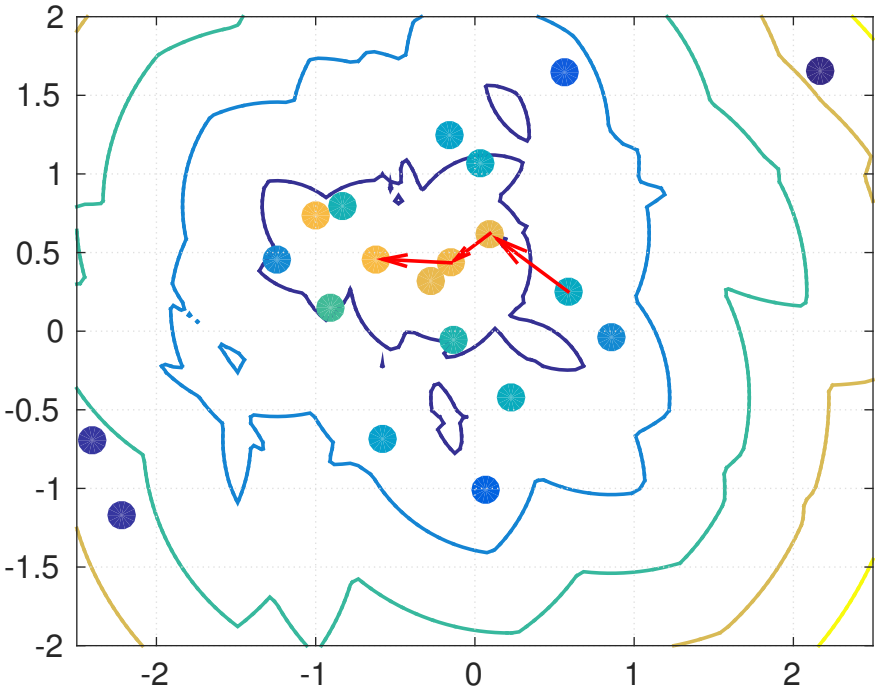


Figure 2: Illustration of the kNN mode seeking algorithm for a standard normal sample of size 20 and $k = 3$. The red arrows shows the trajectory for a single point. The data points are marked with color corresponding to the kNN density value. For better visualization the contour represents the inverse density values (pairwise distance) shown for the entire support of the kNN density.

of the kNN density are also constrained to be represented by data points available in the input set. This is similar in construction to a *medoid* [44]. Duin et al. [29] showed promising benefits in terms of speed, accuracy and robustness when comparing to ordinary mean shift.

Given a kNN-density estimate, where the density at a point $\mathbf{x}$ is the reciprocal of the squared distance to the k-th nearest neighbor $\mathbf{x}_k$:

$$\hat{f}_{kNN}(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{x}_k\|^2}, \tag{5}$$

the kNN mode seeking algorithm can be stated as follows:

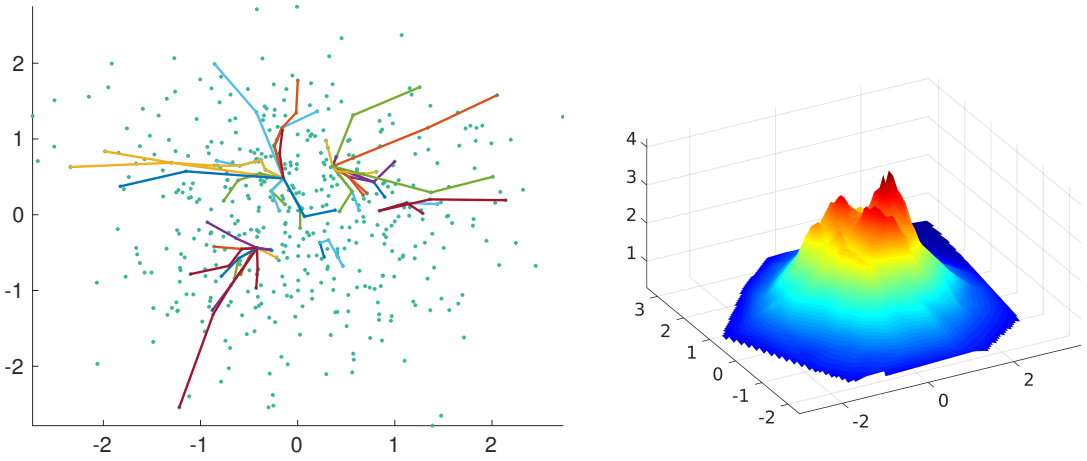
1. For each input point $\mathbf{x}_i$:
   - Define a pointer to the point within the k-nearest neighbors of $\mathbf{x}_i$ with the highest kNN-density.
   - Repeat the process by following pointers from the initial pointer until a pointer that points to itself is found. This will be taken as the local mode of $\hat{f}_{kNN}(\mathbf{x})$.
2. Assign each point, $\mathbf{x}_i$, that converged to the same mode to the same cluster.

This is an approximation of the gradient ascent scheme of mean shift, the pointer to the neighbor with highest density value represents the gradient while a point that points to itself

represents a mode. An illustration of how the algorithm works for a single data point is shown in Figure 2.

This method is significantly faster than mean shift and has comparable accuracy despite only using input points for projection to the mode [29]. The gain in speed comes from the fact that since the trajectories are only input points, the distance matrix that forms the density estimate has to be calculated only once. For mean shift, the matrix has to be updated for each iteration. In addition, as opposed to for example k-means, [30], the method still retains the local properties of mean shift making it adapt closer to the data and less sensitive to the shape of the cluster.

In Figure 3b we see the kNN mode seeking algorithm run on the same data set as in Figure 1. To enable fair comparison we used $k = 20$ neighbors, recall that the mean shift bandwidth was set to the average distance to the 20th neighbor. We see that the kNN density is a very coarse (not smooth) compared to the kernel density example from Figure 1 and we note that the density contains many local maxima even though the sample is from a unimodal density. If we increase the neighborhood size to 100, we get a unimodal density (not shown here).



(a) Trajectories of the kNN mode seeking algorithm.

(b) kNN density.

Figure 3: kNN mode seeking trajectories with $k = 20$ on a random normal sample. The trajectories are shown in random colors to help the reader visually separate them. We note that in this example the kNN mode seeking algorithm found five modes.

To sum up, the kNN mode seeking algorithm is a fast, but crude version of mean shift that scales better to high-dimensional data sets. It inherits most of the other benefits of mean shift, but also the dependence of a critical parameter, in this case the neighborhood parameter k.

4. Proposed method: Parameter free mode based clustering

In this section we present our proposed method: parameter free mode seeking clustering. We achieve this using *consensus clustering*. The core idea of consensus clustering is that

points that are clustered together repeatedly under different cluster settings (parameters, initialization or resampling) or algorithms should be similar [45, 46, 47, 48, 35]. In most cases, consensus clustering consists of a two stage clustering process, which can be summarized as follows:

1. Run multiple clusterings of the data with different parameters, initializations and/or random subsets each time. We will call this part the *kNN mode seeking ensemble*.
2. Combine the results to measure the *consensus* over all the repeated clusterings.

Another main motivation for introducing consensus clustering, is to acknowledge that there is no single clustering algorithm which will be appropriate for every data set and that different algorithms might produce different partitions for the same data set. Even when applying a single clustering algorithm several times to the same data set with different initial conditions or parameters, ambiguous results might arise when outputs are compared. This can often make the interpretation of a clustering result a challenge. Further interesting views can be found in [4].

We start by presenting how the kNN mode seeking ensemble is built in the next subsection and then move on to explain how the consensus over the repeated clusterings is calculated.

4.1. kNN mode seeking ensemble

The goal of a clustering ensemble is robustness, diversity and stability, which is achieved through the natural variation in the data set and the variation of the clustering algorithm induced by random initialization or stochastic optimization [33, 2].

An issue that comes up when building an ensemble using kNN mode seeking is that once the critical parameter of a density estimate is set – the kth nearest neighbor for the kNN density – further operations on the density are deterministic. As one of the key elements in consensus clustering is randomness in either initialization or parameter selection, see for example [49], we need to introduce randomness when using mode seeking in a clustering ensemble.

In this work we have chosen to introduce randomness in both parameter selection and the data set itself. We have used the following strategies to achieve this:

- Random parameter selection: For each repeated clustering the k nearest neighborhood parameter is sampled uniformly from $K = \{k_1, k_2, \cdots, k_n\}$, $k_i \in \mathbb{Z}$.
- Subsampling: We use different subsets of the data for each run of the kNN mode seeking algorithm. The subsampling is performed without replacement.

The subsampling procedure is introduced to add diversity and prevent overfitting. This can also be interpreted as to mimic the random initializations of non-deterministic algorithm to further enhance the ability to capture structure in the data. As Monti et al. [33] states: "Perturbations of the original data can be simulated by resampling techniques". Also, the variation of the k-parameter captures neighborhoods of different size, such that diversity in the data and variation in scale is captured when it is run multiple times.

They are not bound to a particular shape – like e.g. k-means, they are geometrically intuitive and the number of clusters is determined automatically by the algorithm [3].

4.2. Calculating consensus

To evaluate the consensus of the clustering ensemble, we follow Monti et al. [33] and Fred and Jain [32, 34]. We calculate a similarity measure across the individual clusterings, which is averaged to form a similarity matrix. Many methods can be used in the final step. We use hierarchical clustering [30], further explained below, as it is a well established method that has shown promising results in ensemble clustering [50].

Given a data set X to be clustered and M repeated clusterings, we calculate the *consensus matrix*, $\mathcal{C}$, as :

$$\mathcal{C}_{ij} = \frac{S_{ij}}{I_{ij}}, \tag{6}$$

where S_{ij} is the number of times $\mathbf{x}_i$ and $\mathbf{x}_j$ has been assigned to the same cluster. I_{ij} is the number of times $\mathbf{x}_i$ and $\mathbf{x}_j$ are both included in the same random subsets of the data.

The complete consensus matrix can thus be considered a normalized similarity matrix. If two data points are clustered together in many of the different clustering solutions (closer to one), they are considered more similar than two data points that are not clustered together as often (closer to zero). This similarity measure can then be converted to a dissimilarity matrix and used to obtain a final partitioning/clustering. In previous works hierarchical clustering with the single linkage criteria has been used [34, 32, 30].

$$d_{sl}(X, Y) = \min_{x \in X, y \in Y} \|x - y\|^2. \tag{7}$$

This is perhaps the most intuitive approach as it can be interpreted as cutting the links between points that are clustered together less times than a certain threshold. However, it is well known that the single linkage criteria works best when the cluster structures follows elongated and nonlinear cluster structures [30], and can result in induced artifact clusters even if the data does not contain such structures.

We therefore propose to also use another proximity measure used in hierarchical clustering, *average linkage*:

$$d_{al}(X, Y) = \frac{1}{|X|\,|Y|} \sum_{x \in X} \sum_{y \in Y} \|x - y\|^2. \tag{8}$$

Average linkage is more robust and can avoid elongated artifact clusters that can appear in single linkage. At the same time it requires more computations as the similarity matrix has to be updated at each level in the hierarchy [51].

Once a cluster hierarchy has been established the final clustering is obtained by selecting the cluster level that has the longest *lifetime*, where the lifetime of a cluster is defined as the absolute difference of the proximity level at which it is created and the proximity level at which it is merged into a larger cluster [30]. This will automatically return the number of clusters in the data set.

We summarize the proposed algorithm in Algorithm 1.

Algorithm 1 Consensus clustering using kNN mode seeking

Input Data set X, range of k-values K, subsampling rate p and number of clustering trials M.

1: Initialize I and S as $\mathbf{0}_{N\times N}$
2: **for** each clustering trial **do**
3: Draw a random k^* from K.
4: Draw a random sample of size pN, X^*, from X.
5: For each pair of data points $\mathbf{x}_i$, $\mathbf{x}_j$ contained in X^*, $I_{ij} \leftarrow I_{ij} + 1$.
6: Use kNN mode seeking with parameter k^* to obtain a clustering of X^*.
7: For each pair of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in X^* that belong to the same cluster, update S by $S_{ij} = S_{ij} + 1$.
8: **end for**
9: Normalize the concensus matrix, $\mathcal{C}$, by dividing elementwise by the counter matrix; $\mathcal{C}_{ij} = \frac{S_{ij}}{I_{ij}}$
10: Create a dendrogram using hierarchical clustering.
11: Obtain the final clustering by selecting the cluster configuration with the longest lifetime.

Output Clustering C of X.

Synthetic data sets in $\mathbb{R}^2$	Real data
Noisy circles	MNIST (images)
Scale	COIL-20 (images)
Banana ball	Classic 3 (text)
	EHR data (text)

Table 1: Summary of data sets used in the experiments.

5. Results and discussion

In this section we present numerical experiments on both real and synthetic data sets. The synthetic point sets are included to illustrate that the clustering algorithm is able to handle general issues of unsupervised learning such as nonlinearity, difference in scale, clusters that are close in proximity and sensitivity to outliers. Also, these data sets enables us to visually confirm if the clustering results are intuitive or not.

The real data sets are the MNIST handwritten digit images, the Classic 3 collection of online abstracts, the COIL-20 data set in addition to a data set consisting of electronic health records (EHR) for patients at the University Hospital of Northern Norway. A summary of the data sets is presented in Table 1

The health-care data (Section 5.8) consists of free text and we do not have ground truth available for validation purposes. We therefore analyze it qualitatively using wordclouds and a visualization algorithm.

5.1. Experimental setup

To evaluate clustering results, we have used the adjusted rand index (ARI) [52]. This is a standard way of measuring the perfomance of a clustering algorithm by measuring the accuracy of the clustering result compared to a priori ground truth and adjusting for the probability of clustering by chance.

We start with presenting clustering performance in the next section and then proceed to analyze the proposed algorithm in terms of speed. The focus of our contribution is to enhance clustering by mode seeking, therefore we compare the proposed method to standard mean shift and a single run of kNN mode seeking.

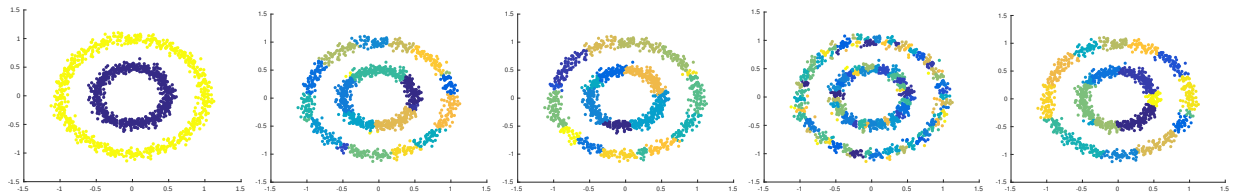
The main focus of this paper is to add robustness to the concept of clustering by mode seeking. We have therefore chosen to compare the proposed algorithm with standard mean shift and a single run of kNN mode seeking. In addition to this we have included several other experiments, both to ensure a proper comparison with modern state-of-the-art ensemble clustering methods as well as emphasising the strength of adding the ensemble In Section 5.9 we have compared our method to spectral ensemble clustering and the Bayesian cluster ensemble. These are methods based on the same consensus matrix strategy, but the last clustering stage is different. Section 5.10 compares our algorithm to a single run of mean shift and kNN mode seeking followed by a stage of hierarchical clustering.

Finally, for completeness we have in Appendix A compared the performance of our algorithm with the original algorithm of Fred and Jain [50]. The mean shift bandwidth parameter, σ^2, is selected as the mean distance to the 10th, 50th and 100th nearest neighbor. In kNN mode seeking we use neighborhood parameter $k = 10,\ 50,\ 100$. These choices will represent a wide range of parameter selections and reflect the overall performance of the algorithms.

Our experience is that the proposed algorithm works for a wide range of parameters. However, for all experiments we used the same parameters, inspired by and extended from the previous work [1], where we used a fixed range of neighborhood parameters to capture cluster structures on different number of scales. This showed promising results, and in this work we use a slightly modified range when sampling the k parameters used in the proposed algorithm:

- Neighborhood parameter k sampled uniformly from $K = \{5, 6, 7, 8, 9, 10\}$.
- Subsampling rate without replacement $p = 80\%$.
- Number of iterations (repeated clusterings) $M = 300$.

In the experiments performed on low-dimensional data sets we observed that average linkage did not give clear dendrograms with respect to lifetime and the correct number of clusters. This is most likely a combination of the chaining problem in single linkage, see e.g. [30], and the fact that in higher dimensions distance measures are less robust, leading to the choice of merging clusters based on the closest data points unstable. Further evaluation of single linkage for high dimensional data is deferred to future research. Because of this observation we chose to use single linkage in all examples where the dimensionality is lower than 5, and average linkage in the examples with higher dimensionality.



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

Figure 4: Noisy circles experiment. (a) Consensus clustering using kNN mode seeking. (b) kNN mode seeking algorithm with $k = 10$. (c) kNN mode seeking algorithm with $k = 100$. (d) Mean shift with bandwidth equal to the mean distance to the 10th NN. (e) Mean shift with bandwidth equal to the mean distance to the 100th NN.

5.2. Toy data: Noisy circles

To illustrate the ability of the algorithm to capture nonlinear structure at different scales using the suggested default parameters, we test it on the 'noisy circles' data set[4]. We compare the result of using Algorithm 1 with regular mean shift and two instances of kNN mode seeking. Visual results comparing with the highest and lowest parameter selections for mean shift and kNN mode seeking are shown in Figure 4. It is evident that the pure mode seeking algorithms can not handle the nonlinearities represented by the two noisy circles. They instead form a range of subclusters that represents local variation, but not the global clustering structure.

Comment: Consider the following: If the data are distributed uniformly along the circles with low variance and zero mean Gaussian noise we will by Definition 1 have a connected mode along each circle. In practice, with unevenly sampled points along the circles, due to the Gaussian noise, we will get local modes and it will thus be impossible for a pure mode seeking algorithm to capture the global cluster structure.

The kNN mode seeking consensus clustering algorithm gives a correct clustering result. In Figure 5 the dendrogram is shown and we clearly see that there are two clusters according to the longest lifetime criteria.

5.3. Toy data: Scale experiment

Cluster structures of varying scale is often a problem for clustering algorithms. We test our algorithm on a toy data set consisting of a combination of linear and nonlinear clusters to illustrate its capability in such situations. Figure 6 shows the results on the scale data set for the proposed algorithm, kNN mode seeking with $k = 50$ and mean shift with σ^2 equal to the average distance to the 100th nearest neighbor. Table 2 shows the ARI score for all the bandwidth selectors. kNN mode seeking finds the correct number of modes at $k = 50$, but the ARI score reveals that some nonlinearities are not captured. Mean shift cannot find the correct number of clusters using any of the selected bandwidth rules, but achieves the highest ARI with 21 clusters. The proposed algorithm achieves a correct clustering result.

[4] `http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html`.

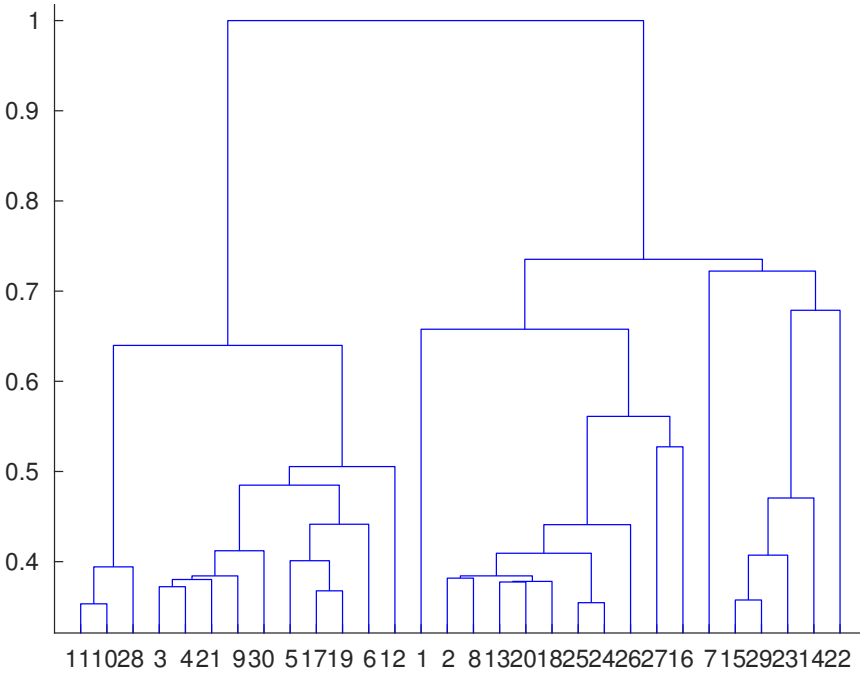


Figure 5: Dendrogram for the noisy circles experiment using the proposed algorithm.

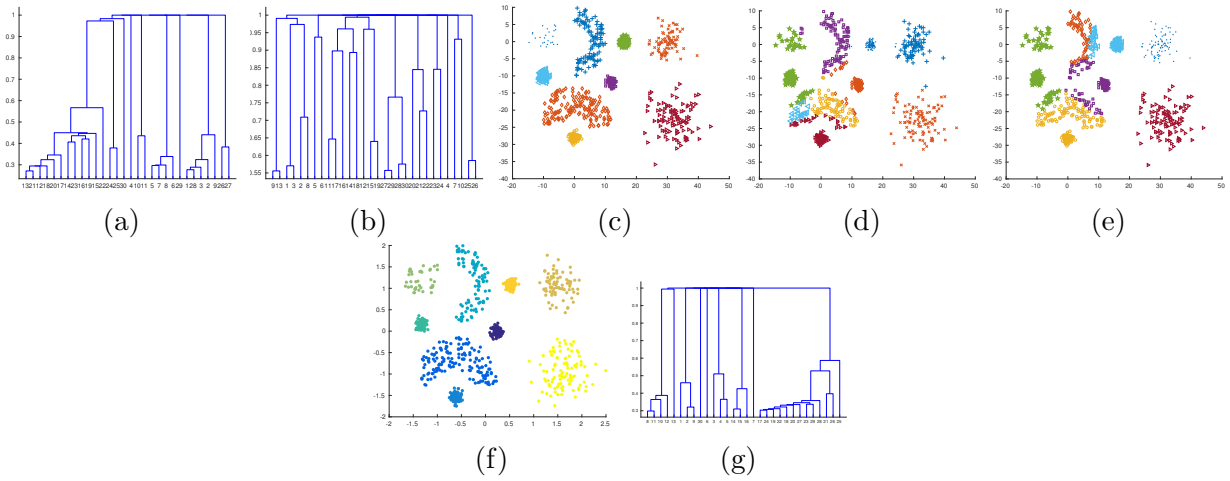


Figure 6: Toy data experiment with clusters of different shape, scale and structure. (a) Single linkage dendrogram. (b) Average linkage dendrogram. (c) Proposed method, using the single linkage criteria. (d) kNN mode seeking with $k = 50$. (e) Mean shift with kernel bandwidth equal to average distance to 100th NN.

Method	k/σ^2	# clusters	ARI
EC_{kNN-MS}	–	9	1
EC_{MS}	–	9	1
kNN-MS	10	28	0.549
	50	9	0.768
	100	4	0.376
MS	1.58	102	0.612
	4.02	21	0.806
	8.96	7	0.688

Table 2: Clustering results, measured by adjusted rand index, on the 9 class toy data set. EC_{kNN-MS} refers to the proposed method, kNN-MSto kNN mode seeking and MSto mean shift.

5.4. Toy data: Banana ball

In the final toy example we mix two nonlinear "banana shaped" clusters with a Gaussian blob, and keep the distance between the clusters low. This data set represents challenges both due to the nonlinear structure and different scales across shapes. Also, all three shapes are set very close to each other, which makes it a hard clustering problem due to the fact that distances within clusters are, for some data points, larger than distances across clusters.

In this example, we compare the proposed algorithm to kNN mode seeking, mean shift *tuned to return the correct number of clusters*, as well as the kNN mode seeking with a low number of neighbors. This is to further illustrate the robustness of the proposed framework compared to the sensitivity of the pure mode seeking algorithms to bandwidth parameter selection. The results are shown in Figure 7. From the the subfigures of Figure 7 we again see that the pure mode seeking algorithms cannot correctly cluster the nonlinear shapes. The mean shift algorithm seems to perform better as the Gaussian blob is cleanly separated from the rest, but the nonlinear structures are not correctly clustered. In Figure 7c we see that the proposed algorithm correctly clusters all three classes except a few points. The dendrogram is shown in Figure 8 and we see that the lifetime criteria gives three clusters. This results in an ARI score of 0.9974.

In Figure 7d a neighborhood parameter of $k = 15$ was used as an example and we see that the kNN mode seeking captures many local structures *within each cluster*. Figure 9 shows a 3D plot of the corresponding estimated kNN density ($k = 15$). This figure contains an important observation; the global cluster structures are represented by elevated density regions, but at the same time they are completely dominated by local maxima that will clearly upset the kNN mode seeking result. We see that the proposed algorithm is able to capture the elevated regions and obtain a coherent clustering result despite the problematic local maxima.

To summarize the experiments performed on the synthetic data sets we see that the proposed algorithm with the default parameters is able to capture global cluster structures containing local maxima as well as nonlinear shapes and clusters that are close in proximity. Other mode seeking algorithms struggle with these issues, especially when it comes to capturing global cluster structure and clusters that are close in proximity.

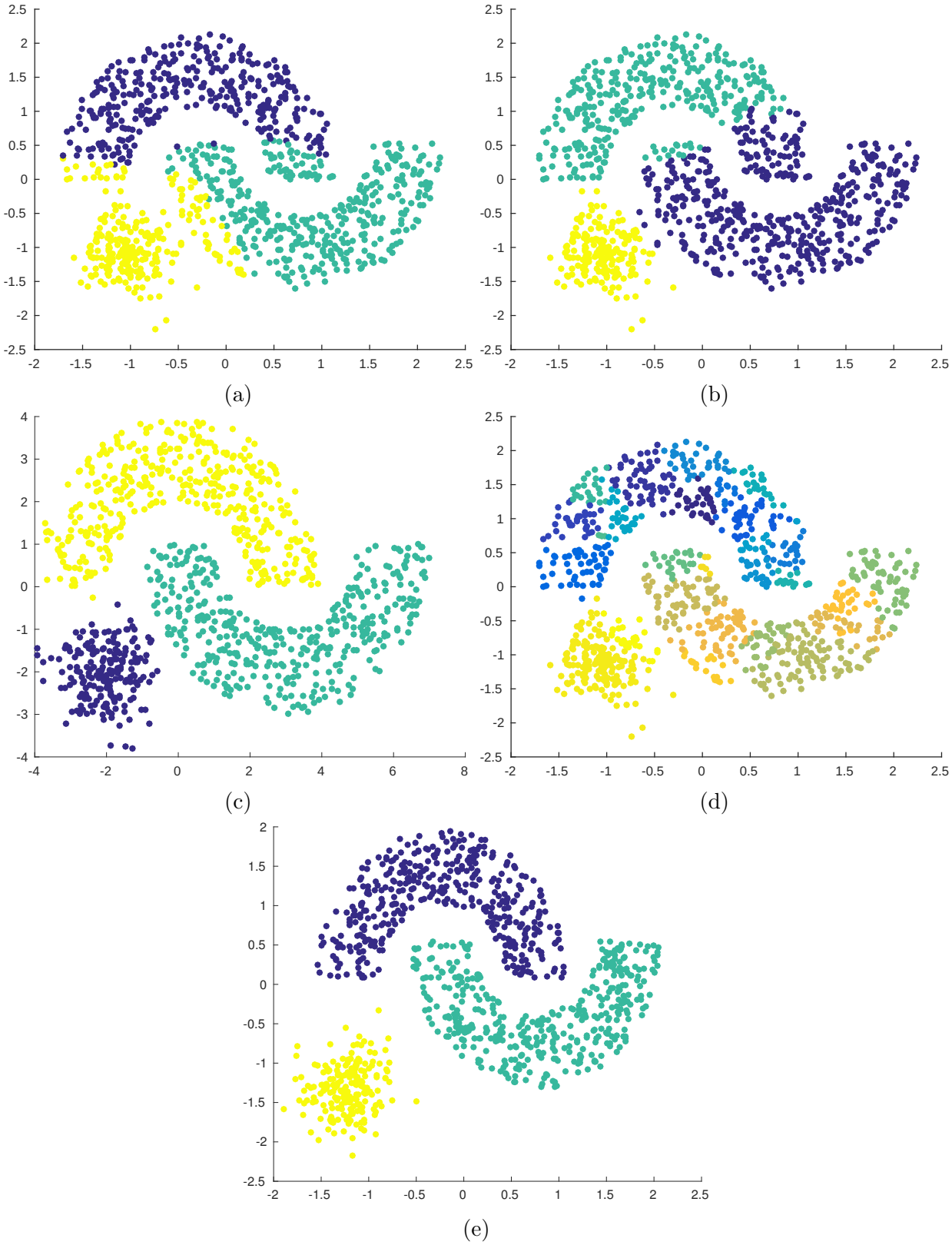


Figure 7: Banana ball experiment. (a) kNN mode seeking with $k = 120$. This selection of k returns three clusters. (b) Mean shift with $\sigma = 1.0$, which returns three clusters. (c) Results of the proposed algorithm. (d) kNN mode seeking with $k = 15$.

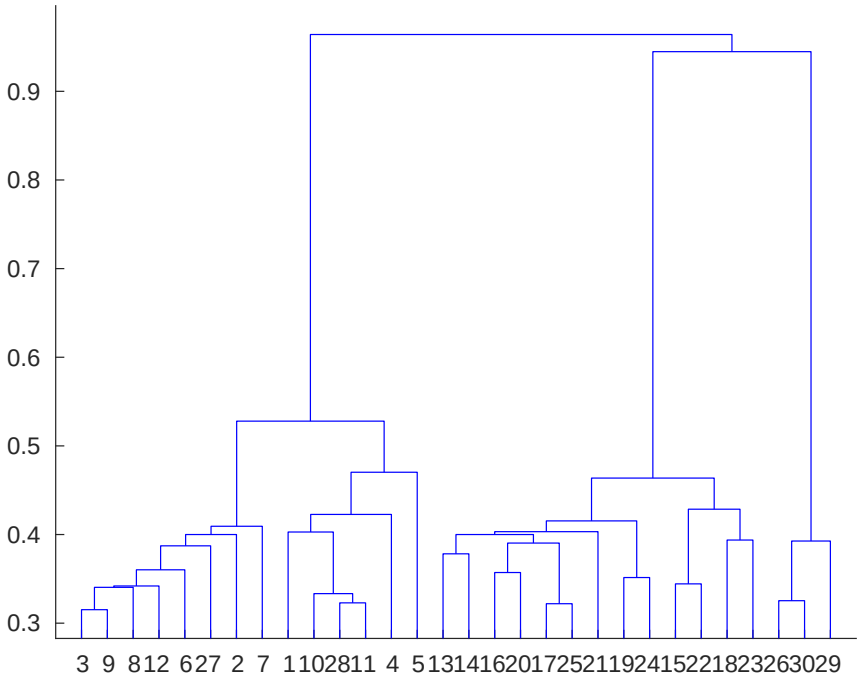


Figure 8: Dendrogram for the banana ball experiments using the proposed algorithm. We see that cutting the dengrogram based on the longest lasting cluster configuration clearly gives three clusters, which is the correct choice.

5.5. MNIST digit images

We evaluate the proposed clustering algorithm on the MNIST handwritten digits [53]. We used the training data consisting of 10000 digits (approximately 1000 of each) with vectorized pixel values as features. No further preprocessing or feature extraction was done and the previously suggested default parameters was used. Table 3 shows the results of the proposed method, kNN mode seeking and mean shift with the same bandwidth selectors as previously used. The average linkage dendrogram is shown in Figure 10 where we see that the longest lifetime gives 10 clusters which is the correct choice. As a further validity check we run the k-means algorithm, which is perhaps the most used clustering algorithm, with 10 clusters [30] as input. The k-means clustering results in an ARI of .3367, a clear indicator that the MNIST digits can not be described by spherical clusters that are linearly separable.

As seen, our proposed algorithm outperforms the other mode seeking algorithms as well as k-means without parameter tuning and returns the correct number of clusters when using the average linkage dendrogram.

5.6. Coil-20 data set

We run our algorithm on the Coil-20 data set [54] with default parameters and compare with kNN mode seeking and mean shift. It is a data set consisting of images of 20 different objects taken at several different angles, resulting in highly nonlinear structure within each image. The dimensionality is 16384, so the high-dimensional performance of the algorithm is

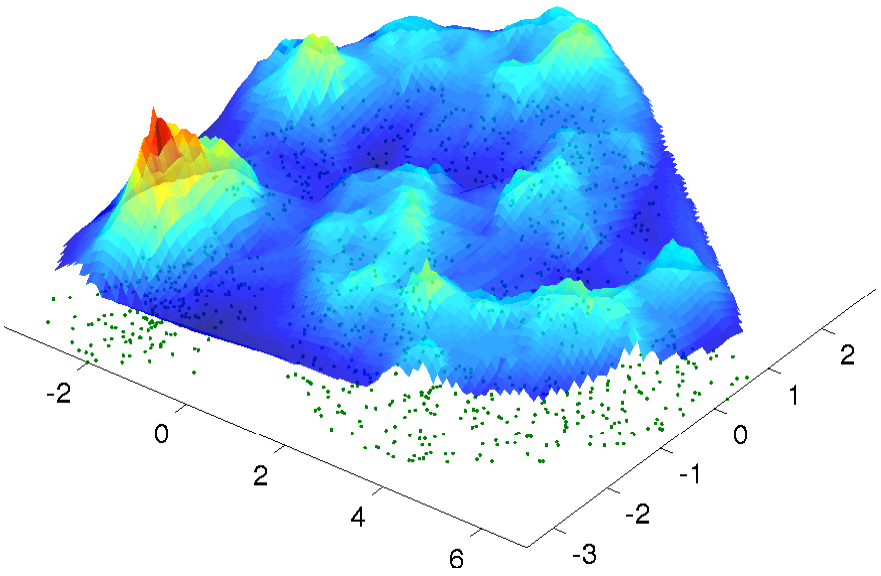


Figure 9: 3D plot of the kNN density estimate with $k = 15$ for the banana ball density. The data set is shown on the XY axis(the two horizontal axes), and the a surface plot of the density is on the Z axis(the vertical axis)

Method	k/σ^2	# clusters	ARI
EC_{kNN-MS}	–	–	0.5850
EC_{MS}	–	–	0.00002
kNN-MS	10	112	0.1795
	50	18	0.3584
	100	9	0.2424
MS	10th	1558	0.2437
	50th	383	0.0435
	100th	160	0.0392

Table 3: Clustering results, as measured by adjusted rand index, on the MNIST data set. The σ^2 parameter is set as the average distance to the given neighbor.

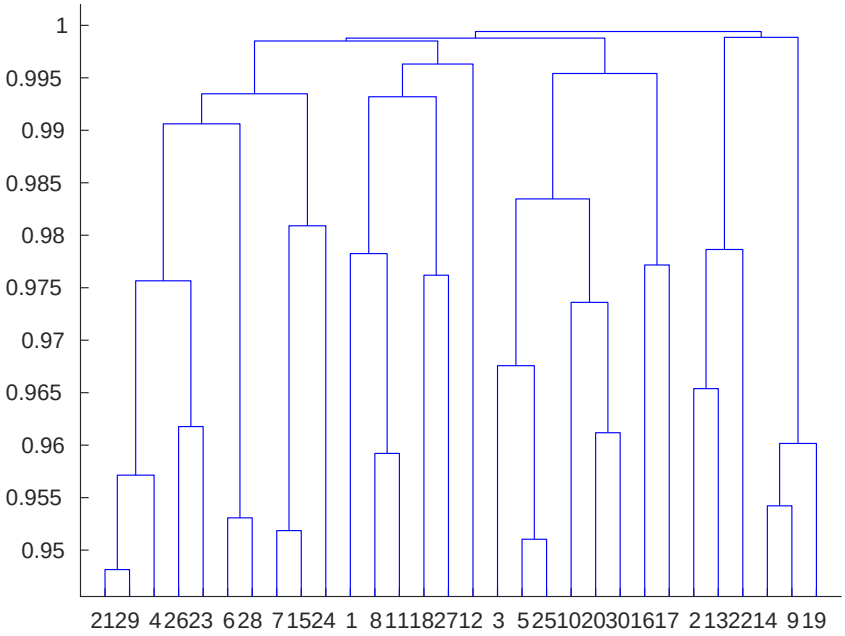


Figure 10: Dendrogram for the full MNIST 10000 digits.

d	EC_{kNN-MS}	kNN-MS	MS
500	.5487	.5644	.5124
16384	.5730 (31s)	.3440 (3s)	.5057 (8823s)

Table 4: Clustering results, measured by adjusted rand index, on the COIL-20 data set. The computation times for the full dimensional data set is included in parentheses.

crucial. As the concepts of nearest neighbors and average distances breaks down in such high dimensionality, we perform a parameter sweep over $k = 5 : 100$ for the kNN mode seeking and 100 values chosen linearly in the interval between the largest and smallest pairwise distances of the data for the mean shift algorithm. To illustrate the effectiveness of our proposed algorithm in higher dimensionality we run the experiment on both the full version of the COIL-20 data set as well as a version where the dimensionality has been reduced to 500 using principal component analysis. The results are shown in Table 4. Our method has the highest ARI and it works in the full dimensional space. We notice that the results of the mean shift algorithm is comparable to our algorithm, but when the time factor is taken into account the benefits of the proposed method is quite clear. Mean shift with the kernel bandwidth that gives the highest ARI score for the full dimensional results takes about 8823 seconds to run, while the proposed algorithm runs in approximately 31 seconds[5]. The kNN mode seeking algorithm runs in approximately 3 seconds in the full dimensional case, which is by far the fastest, but the low ARI score shows that the increase in speed results in reduced accuracy.

[5]This renders a mean shift based ensemble impossible to use in practice.

Method	k/σ^2	# clusters	ARI
EC_{kNN-MS}	–	3	0.9506
EC_{MS}	–	3	0.0003
kNN-MS	10	76	0.1167
	50	7	0.4050
	100	6	0.3117
MS	1.32	1	0
	1	74	0.0048
	0.8	3848	0.00001

Table 5: Clustering results, measured by adjusted rand index, on the classic 3 data set.

5.7. Classic 3 data set

We evaluate our algorithm on a data set involving text, the Classic 3 data set[6]. It is a collection of abstracts from three online repositories of different journals [55]. We preprocess the data set using a weighted term frequency - inverse document frequency (TF-IDF) scheme [56]. The features are normalized to one and we use the suggested default parameters. The average linkage dendrogram, not shown here, gives three clusters based on the lifetime criteria. Results are summarized in Table 5. For the mean shift algorithm all the suggested bandwidth estimators returned a single cluster, which is a clear indicator of oversmoothing. We tried smaller bandwidths, but the results deteriorated quickly, giving 74 and 3848 clusters which are clearly not meaningful.

5.8. Case study: Patient stratification

We conclude the experiments by including a case study with real data where the proposed clustering method is used. The case study aims to use written text to identify interesting sub-populations related to mental well-being in a cohort consisting of patients that have undergone major surgery. The written text is represented by nurses notes from the patients electronic health record (EHR) in a period of 20 days after the surgery. We extracted EHRs for 1138 patients that had undergone a major surgery from the department of gastrointestinal surgery at the University Hospital of North-Norway [57]. The final data set consists of the top 15 principal components of the term frequency - inverse document frequency [58] for each patient[7].

We applied the proposed clustering algorithm using the default parameters presented earlier and based on the dendrogram, shown in Figure 13a, the number of clusters was found to be 5 based on the longest cluster lifetime. As we have no labelled ground truth for the patient data we evaluate the results qualitatively using visualization with t-SNE [59] and wordclouds [60].

[6] `https://sites.google.com/site/fawadsyed/datasets`

[7] Each patient has a collection of documents that are concatenated together to form one string of text for each patient.

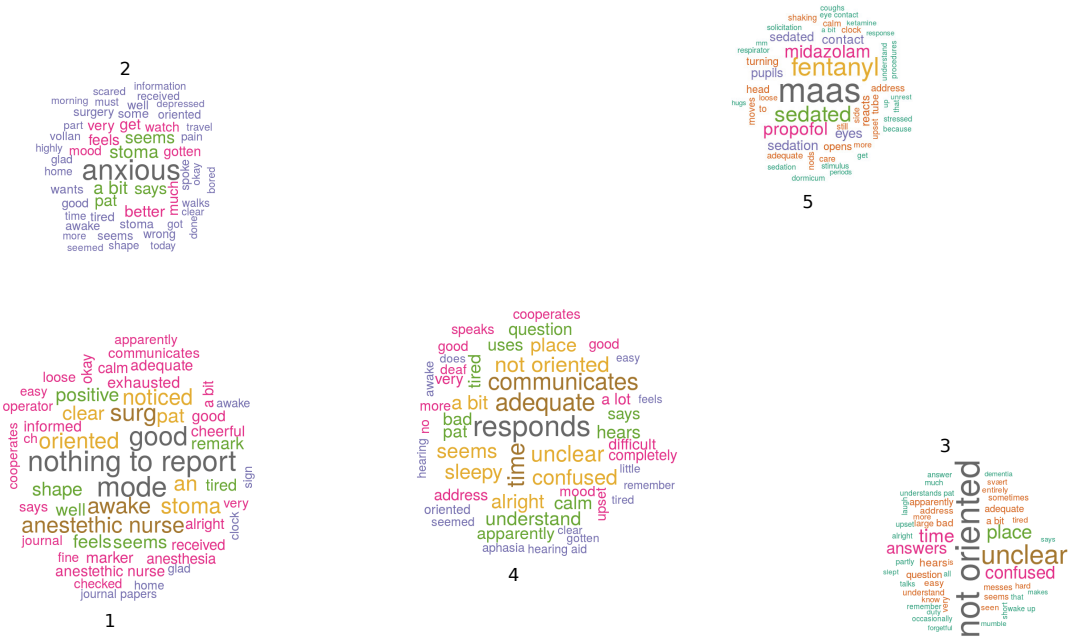


Figure 11: Wordclouds that illustrate the most frequent features (words) in the 5 clusters automatically found by the clustering algorithm. The figure also illustrates the distances between the cluster means. Number of patients in each cluster: bottom left (cluster 1): 684, upper left (2): 248, center (4): 140, upper right (5): 32, bottom right (3): 34.

First of all we note that the clusters vary a lot in size, from 32 to 684 patients. Table 6 provides a qualitative summary of the clustering results. In Figure 11 we have plotted wordclouds for each individual cluster in order to illustrate the most common words and overall theme of each cluster. The figure also illustrates the (approximate) distances between the cluster means (Euclidean distances on PCA features). Cluster 1, which by far is the largest cluster, seems to contain "ordinary" patients with normal, positive outcomes. Frequent words/phrases are *good mood, positive, awake, nothing to report, oriented, clear*. Cluster 2 also contains seemingly normal patients, but interpreting the common words suggests that these patients are more anxious and worried. The, by far, most frequent word is *anxious*. The two smallest clusters (cluster 3 and 5) contain specific patient cohorts, in cluster 3 words like *confused*, *unclear*, *not oriented* dominate, whereas in cluster 5 the theme appears to be sedation and related drugs. Frequent words are *maas* (motor activity assessment score, used to measure pain), *fentanyl, propofol, midazolam* (sedation drugs). Cluster 4 is placed between the normal cluster (cluster 1) and the cluster with confused patients (cluster 3). The words *confused*, *unclear* and *not oriented* are quite frequent here, but they cannot be

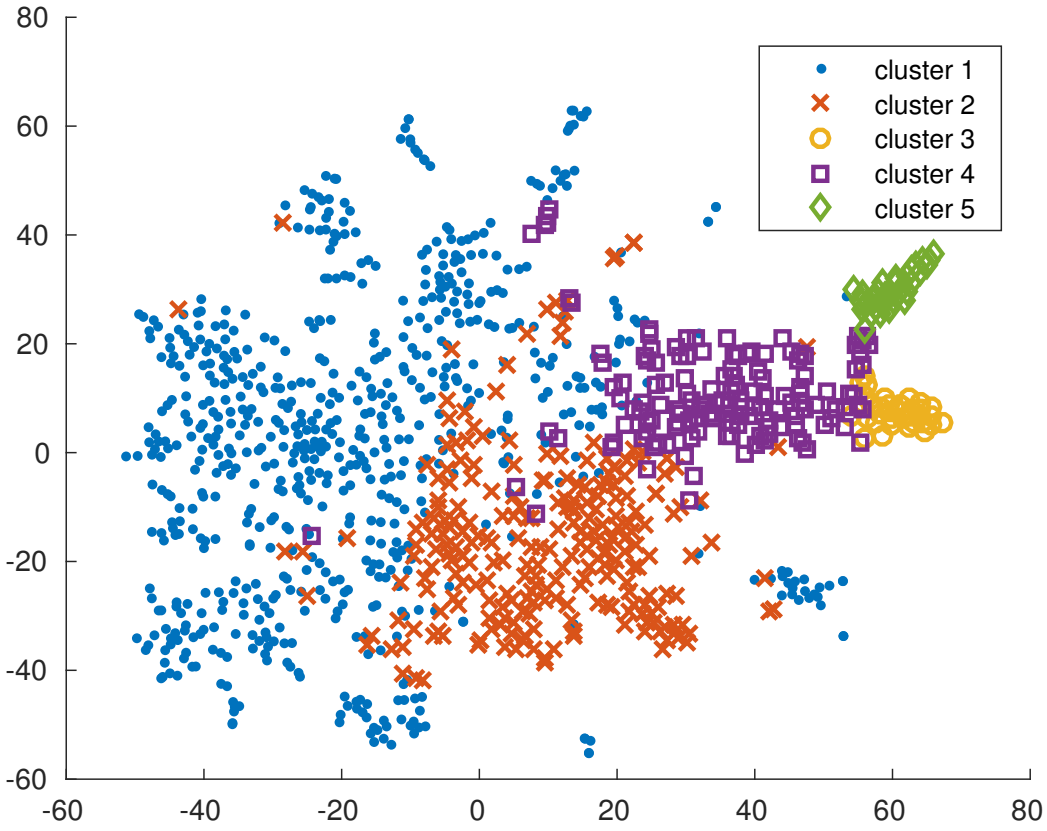


Figure 12: EHR data set: Dimensionality reduced to two using t-SNE. The labels from clustering with the proposed algorithm shown in different colors and symbols.

said to dominate. The most frequent words are related to speech and communication.

To further analyze the results the clustered data points are visualized in two dimensions using t-SNE, [59], shown in Figure 12. The color coding and different symbols indicates cluster labels. We see that the cluster labels represent compact and visually intuitive clusters for all but the blue cluster (cluster 2). The green, brown and yellow clusters (cluster 3, 4 and 5) forms a group that is separated from the rest and has similar themes when the word clouds are inspected.

Upon reviewing the wordclouds and the low dimensional visualization results we find the results both intuitive and promising

When we run ordinary mean shift on the data set we get an unrealistically high number of clusters (> 150) using the rule-of-thumb bandwidths. We manually tuned the bandwidth such that the algorithms outputs 5 clusters. The result is visualized in Figure 13b. Mean shift gives in this case one large cluster and four small clusters with one or two members each, a result that is very hard to interpret further.

The result obtained by running a kNN mode seeking with $k = 10$ is shown in Figure 13d.

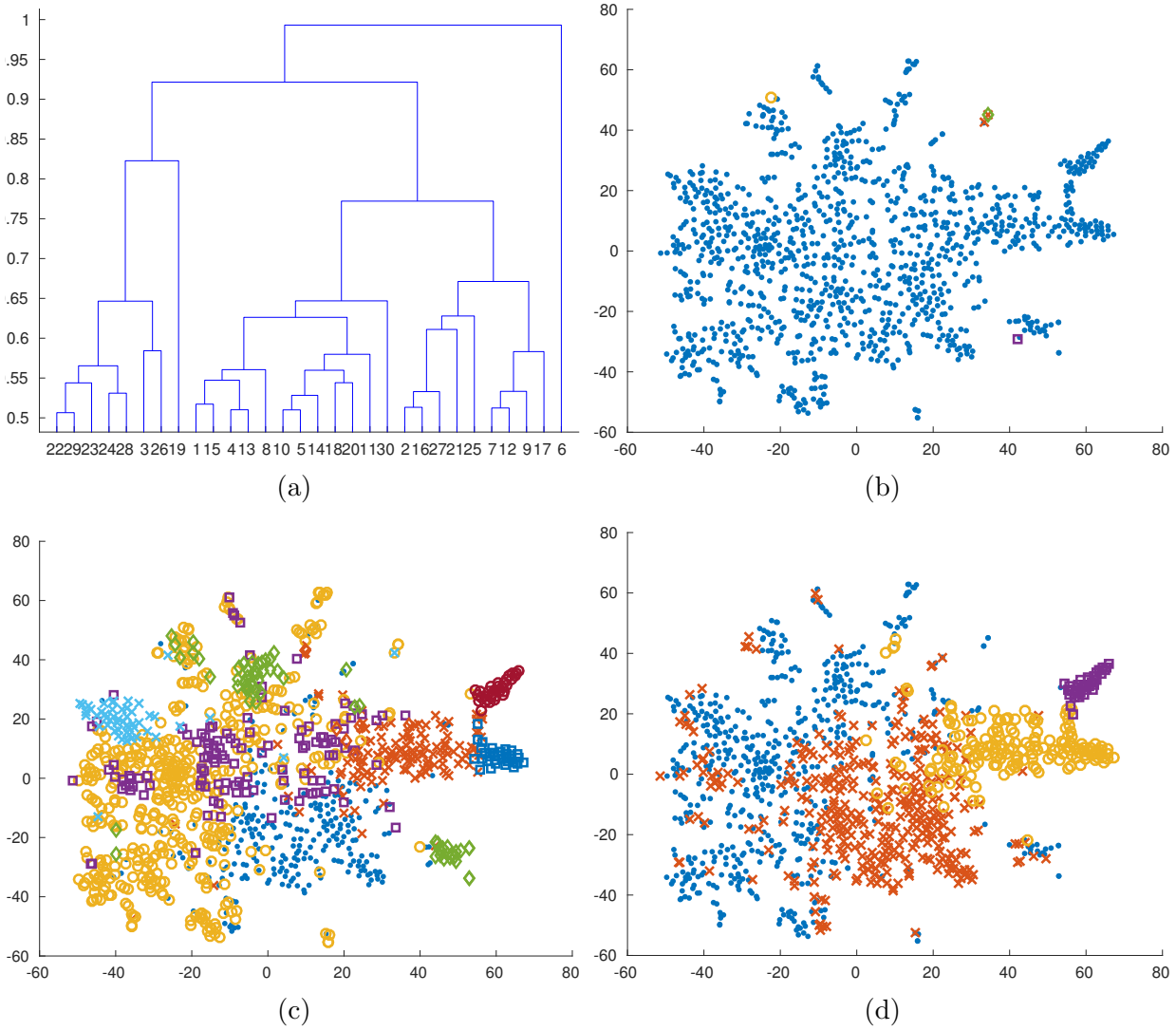


Figure 13: Patient stratification case study. (a) Average linkage dendrogram. (b) Mean shift. (c) kNN mode seeking with $k = 8$. (d) knn mode seeking with $k = 10$.

	# of patients	Keywords and themes
•	684	*Good mood* and *nothing to report*
×	248	*Worried*
○	34	*Not oriented* and *confused*
□	140	*Adequate* and *communicates*
◇	32	*Sedation* and *sedation drugs*

Table 6: Summary of clustering results on the EHR data set. The table shows the number of patients that belong to each cluster, the marker and color that represents the cluster in the t-SNE maps and certain keywords that describe the different clusters.

Dataset \Clustering method	K means	kNN mode seeking	Mean shift	EC_{kNN-MS}
Banana ball	0.9896	0.9922	0.9948	0.9974
Scale	1	0.5871	1	1
Classic 3	0.9825	0.9324	-	0.9506
COIL-20	0.6596	0.5346	0.2635	0.5730
MNIST	0.3756	0.5847	-	0.5850

Table 7: Spectral ensemble clustering results using different clustering methods for building the ensemble.

$k = 50$ and $k = 100$ results in a single large cluster. Figure 13c shows the output from kNN mode seeking using $k = 8$. We see that reducing the neighborhood size from 10 to 8 has a severe impact on the final result. This is included to illustrate that the kNN mode seeking algorithm is very sensitive to the parameter choice, whereas the proposed method is much more robust. It can also serve as a possible explanation to why combining several kNN mode seeking runs gives better results. Comparing the two results we see that they capture very different structures, smaller clusters at $k = 8$ and slightly larger structures at $k = 10$. If we then compare with the result of the proposed algorithm where the kNN ensemble has been used, we can see cluster structures that are evident in both the single run results.

5.9. Comparison with state of the art algorithms

In this section we compare the proposed algorithm to two state-of-the-art consensus clustering algorithm, *Spectral Ensemble Clustering* (SEC)by Liu et al. [61] and *Bayesian Cluster Ensembles* (BCE) by Wang et al. [62]. These algorithms are similar in construction to our method as they are based on the same co-association stage, but in the final clustering they differ.

SEC is simply a k-means based consensus matrix, constructed in the same way as done in this work, followed by a stage of spectral clustering. BCE adds a Bayesian twist to the consensus stage by assuming that the initial clusterings is a sample from a Dirichlet distribution and propose variational methods to perform the final clustering. Further details are beyond the scope of this paper, furher details can be found in [62] and [61].

Both algorithm suggest initial parameters, which were used exclusively in the experiments. Also, we note that in both methods, the final number of clusters returned by the algorithm has to be given as input.

We ran both algorithms on all the datasets except the noisy circles and the EHR data. Results are shown in Table 7 and Table 8

5.10. Separating the effect of the ensemble from the effect of the hierarchical clustering after mode seeking

To illustrate that the consensus stage/clustering ensemble is in fact the key ingredient for adding robustness, we compare to the alternative setup of a single run of mode seeking followed by hierarchical clustering.

Dataset	Bayesian consensus clustering
Banana ball	.4074
Scale	.7293
Classic 3	.9086
COIL-20	5807
MNIST	.3896

Table 8: Bayesian cluster ensemble

Dataset	kNN mode seeking	Mean shift	Linkage method
Banana ball	0.9974	0.6829	(Single link)
	0.5248	0.4428	(Average link)
Scale	0.8963	0.8374	(Single link)
	0.8642	0.1248	(Average link)
Classic 3	0.5524	-	(Single link)
	0.5102	-	(Average link)
COIL-20	0.5524	0.2023	(Single link)
	0.5102	0.5117	(Average link)
MNIST	0.3818	0.4556	(Single link)
	0.2135	0.0333	(Average link)

Table 9: Clustering results (ARI) obtained using single runs of kNN mode seeking and mean shift combined with hierarchical clustering.

We test both kNN mode seeking and mean shift followed by both average link and single link on all three toy datasets, the MNIST digits and the COIL-20 images. Longest lifetime is still used as the final clustering criteria. For simplicity and to get the best possible result from the single stage algorithms, we sweep over a range of σ^2 and k values and select the ones with the best ARI compared to the ground truth. Results are shown in Table 9. In most cases the ARI is lower compared to the proposed algorithm. We also note that the pattern of single link vs average link in high and low dimensions is not as prevalent here.

5.11. Running times: kNN mode seeking vs mean shift

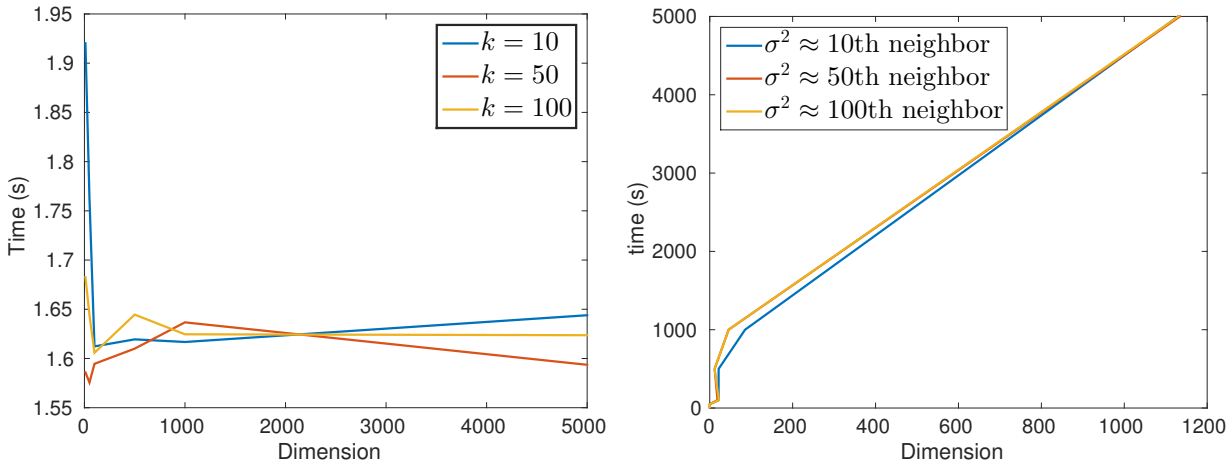
In this section we include two experiments that clearly illustrates the benefits of the kNN mode seeking algorithm in terms of running times. The experiments are run using an Ubuntu 14.04 64-bit system with 64 GB RAM and an Intel Xeon E5-2630 v3 processor. We run mean shift and kNN mode seeking on the MNIST digits data set with the suggested bandwidth selectors and measure the time taken. The algorithms are tested on both the full dimensional version of the data (784 dimensions) and a version reduced to 500 dimensions using principal component analysis. The results are presented in Table 11 and we see that the kNN mode seeking algorithm is much faster compared to mean shift. In the suggested consensus framework this is even more important since the algorithm will be repeated several times. We also note that reducing the dimensionality improves the mean shift algorithm in

Method	d	k		
		10	50	100
MS	784	631.5s	345.4s	202.8s
	500	446.5s	218.2s	118.9s
	10	23.6s	11.3s	7.9s
kNN-MS	784	3.6s	3.0s	3.0s
	500	3.7s	3.5s	3.7s
	10	4.0s	3.9s	3.9s

Table 10: Time taken (in seconds) for clustering the MNIST data set using kNN mode seeking and mean shift. The following bandwidth selectors were used; σ^2 equal to kth average neighbor for mean shift and kth nearest neighbor for kNN mode seeking. The d denotes input dimension, reduced using PCA accordingly.

Method	d	k		
		10	50	100
EC_{kNN-MS}	–	642.2s	–	–
EC_{MS}	–	46121s	–	–
MS	784	152.8s	82.7s	45.2s
	500	103.6s	59.7s	36.9s
	10	10.2s	5.9s	4.04s
kNN-MS	784	3.0s	3.0s	3.1s
	500	2.6s	2.7s	2.8s
	10	2.2s	2.1s	2.1s

Table 11: Time taken (in seconds) for clustering the MNIST data set using kNN mode seeking and mean shift. The proposed bandwidth selectors were used; σ^2 equal to kth average neighbor for mean shift and kth nearest neighbor for kNN mode seeking. The d denotes input dimension, which was reduced using PCA.



(a) (b)

Figure 14: (a) kNN mode seeking on a sample of varying dimensionality from a standard normal distribution. Three different neighborhood sizes was used, 10th, 50th and 100th neighbor. (b) Mean shift on a sample of varying dimensionality from a standard normal distribution. Three different kernel bandwidths were used, average distance to 10th, 50th and 100th neighbor.

terms of speed, but it is still several orders slower compared to kNN mode seeking. The kNN mode seeking actually takes a bit longer time in lower dimensions due to more clusters found by the algorithm, but when comparing the time taken by the full dimensional to the lower dimensional results we see that this factor can be disregarded in practice.

We run both mean shift and kNN mode seeking with the proposed bandwidth selectors on samples from a standard normal distribution with increasing dimension from 2 to 1000 for mean shift and 2 to 5000 for kNN mode seeking. The resulting computation times are shown in Figure 14a and Figure 14b. Comparing to the results on the MNIST, the same trend is evident here: mean shift is increasing in time as a function of dimensionality while kNN mode seeking is close to constant in time as a function of dimensionality.

6. Summary and conclusion

In this paper we have shown that introducing ensemble strategies into mode based clustering can significantly increase the robustness towards parameter selection and high dimensional data. We have suggested default parameters and shown that the proposed algorithm can be used with these to perform exploratory data analysis which shows promising results for both text and image data. The parameters were tested and kept unchanged throughout all experiments, a considerable indicator that the proposed algorithm is robust. In all experiments the lifetime criteria was used, and gave the correct number of clusters in most cases.

We conclude with a list of interesting directions of future research:

- For large scale tasks sparse hierarchical clustering could be used [63, 64].

- The recent robust single linkage by Chaudhuri et al. [65] could replace the hierarchical stage in this paper.
- Spectral clustering techniques could be used in the final step [66, 1, 67]
- Quick Shift or medioid shift could replace kNN mode seeking [68, 69].
- Different ensemble combination strategies should also be investigated [35, 70].

Acknowledgements

This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines*. We would also like to thank the anonymous reviewers for helpful comments.

References

[1] J. N. Myhre, K. Ø. Mikalsen, S. Løkse, R. Jenssen, Consensus clustering using kNN mode seeking, in: Image Analysis: 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15-17, 2015. Proceedings, Springer International Publishing, Cham, 2015, pp. 175–186.
[2] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, International Journal of Pattern Recognition and Artificial Intelligence 25 (03) (2011) 337–372.
[3] G. Menardi, A review on modal clustering, International Statistical Review 84 (3) (2016) 413–433.
[4] I. Guyon, U. Von Luxburg, R. C. Williamson, Clustering: Science or art, in: NIPS 2009 workshop on clustering theory, 2009, pp. 1–11.
[5] A. K. Jain, Data clustering: 50 years beyond k-means, Pattern recognition letters 31 (8) (2010) 651–666.
[6] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM computing surveys (CSUR) 31 (3) (1999) 264–323.
[7] U. Von Luxburg, S. Ben-David, Towards a statistical theory of clustering, in: Pascal workshop on statistics and optimization of clustering, Citeseer, 2005, pp. 20–26.
[8] M. Girolami, Mercer kernel-based clustering in feature space, IEEE Transactions on Neural Networks 13 (3) (2002) 780–784.
[9] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[10] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.
[11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2011, pp. 1297–1304.
[12] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.
[13] J. Agersborg, R. Jenssen, Mean shift spectral clustering using kernel entropy component analysis, in: Proceedings of IEEE Workshop on Machine Learning for Signal Processing, Reims, France, Sept. 21-24, 2014.
[14] U. Ozertem, D. Erdogmus, R. Jenssen, Mean shift spectral clustering, Pattern Recognition 41 (6) (2008) 1924–1938.
[15] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, C. Zoppetti, Nonparametric change detection in multitemporal SAR images based on mean-shift clustering, IEEE Transactions on Geoscience and Remote Sensing 51 (4) (2013) 2022–2031.

[16] J. Böttger, A. Schäfer, G. Lohmann, A. Villringer, D. S. Margulies, Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain, IEEE transactions on visualization and computer graphics 20 (3) (2014) 471–480.
[17] S. Anand, S. Mittal, O. Tuzel, P. Meer, Semi-supervised kernel mean shift clustering, IEEE transactions on pattern analysis and machine intelligence 36 (6) (2014) 1201–1215.
[18] V. Miranda, A. R. G. Castro, S. Lima, Diagnosing faults in power transformers with autoassociative neural networks and mean shift, IEEE Transactions on Power Delivery 27 (3) (2012) 1350–1357.
[19] H. Liu, S. Yan, Robust graph mode seeking by graph shift, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 671–678.
[20] R. Subbarao, P. Meer, Nonlinear mean shift over riemannian manifolds, International Journal of Computer Vision 84 (1) (2009) 1–20.
[21] M. Cho, K. M. Lee, Mode-seeking on graphs via random walks, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 606–613.
[22] Y. A. Sheikh, E. A. Khan, T. Kanade, Mode-seeking by medoidshifts, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.
[23] O. Hyrien, A. Baran, Fast nonparametric density-based clustering of large data sets using a stochastic approximation mean-shift algorithm, Journal of Computational and Graphical Statistics 25 (3) (2016) 899–916.
[24] Y.-C. Chen, C. R. Genovese, L. Wasserman, et al., A comprehensive approach to mode clustering, Electronic Journal of Statistics 10 (1) (2016) 210–241.
[25] J. E. Chacón, P. Monfort, A comparison of bandwidth selectors for mean shift clustering, arXiv preprint arXiv:1310.7855, 2013.
[26] B. Thiesson, J. Kim, Fast variational mode-seeking, in: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, Vol. 22 of Proceedings of Machine Learning Research, PMLR, 2012, pp. 1230–1242.
[27] C. Yang, R. Duraiswami, N. A. Gumerov, L. Davis, Improved fast Gauss transform and efficient kernel density estimation, in: Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, IEEE, 2003, pp. 464–471.
[28] V. C. Raykar, C. Yang, R. Duraiswami, N. Gumerov, Fast computation of sums of Gaussians in high dimensions, UMD–CS–TR–4767, 2005.
[29] R. P. Duin, A. L. Fred, M. Loog, E. Pekalska, Mode seeking clustering by kNN and mean shift evaluated, in: Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2012, pp. 51–59.
[30] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th Edition, Academic Press, San Diego, 2009.
[31] M. Steinbach, L. Ertöz, V. Kumar, The challenges of clustering high dimensional data, in: New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition, Springer Berlin Heidelberg, 2004, pp. 273–309.
[32] A. L. N. Fred, A. K. Jain, Evidence accumulation clustering based on the k-means algorithm, in: Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops SSPR 2002 and SPR 2002 Windsor, Ontario, Canada, August 6–9, 2002 Proceedings, Springer Berlin Heidelberg, 2002, pp. 442–451.
[33] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data, Machine Learning 52 (1-2) (2003) 91–118.
[34] A. L. N. Fred, Finding consistent clusters in data partitions, in: Multiple Classifier Systems: Second International Workshop, MCS 2001 Cambridge, UK, July 2–4, 2001 Proceedings, Springer Berlin Heidelberg, 2001, pp. 309–318.
[35] A. Strehl, J. Ghosh, Cluster ensembles — a knowledge reuse framework for combining multiple partitions, The Journal of Machine Learning Research 3 (2003) 583–617.
[36] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
[37] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall/CRC, 1994.
[38] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and

Machine Intelligence 17 (8) (1995) 790–799.
[39] E. Arias-Castro, D. Mason, B. Pelletier, On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm, Journal of Machine Learning Research 17 (43) (2016) 1–28.
[40] M. P. Wand, M. C. Jones, Kernel smoothing, Crc Press, 1994.
[41] J. E. Chacón, Clusters and water flows: a novel approach to modal clustering through Morse theory, arXiv preprint arXiv:1212.1384, 2012.
[42] S. Ray, B. G. Lindsay, The topography of multivariate normal mixtures, Annals of Statistics (2005) 2042–2065.
[43] W. L. Koontz, P. M. Narendra, K. Fukunaga, A graph-theoretic approach to nonparametric cluster analysis, IEEE Transactions on Computers 25 (9) (1976) 936–944.
[44] V. Estivill-Castrol, A. T. Murray, Discovering associations in spatial data—an efficient medoid based approach, in: Research and Development in Knowledge Discovery and Data Mining: Second Pacific-Asia Conference, PAKDD-98 Melbourne, Australia, April 15–17, 1998 Proceedings, Springer Berlin Heidelberg, 1998, pp. 110–121.
[45] J. Li, S. Ray, B. G. Lindsay, A nonparametric statistical approach to clustering via mode identification, Journal of Machine Learning Research 8 (8) (2007) 1687–1723.
[46] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data, Machine learning 52 (1-2) (2003) 91–118.
[47] A. Lourenço, S. R. Bulò, N. Rebagliati, A. L. Fred, M. A. Figueiredo, M. Pelillo, Probabilistic consensus clustering using evidence accumulation, Machine Learning 98 (1-2) (2015) 331–357.
[48] A. Topchy, A. K. Jain, W. Punch, Clustering ensembles: Models of consensus and weak partitions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (12) (2005) 1866–1881.
[49] A. Goder, V. Filkov, Consensus clustering algorithms: Comparison and refinement, in: Proceedings of the Meeting on Algorithm Engineering & Expermiments, Society for Industrial and Applied Mathematics, 2008, pp. 109–117.
[50] A. L. N. Fred, A. K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 835–850.
[51] H. K. Seifoddini, Single linkage versus average linkage clustering in machine cells formation applications, Computers & Industrial Engineering 16 (3) (1989) 419–426.
[52] D. Steinley, Properties of the Hubert-Arable adjusted rand index, Psychological methods 9 (3) (2004) 386–396.
[53] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
[54] S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library (coil-20), Tech. rep., technical report CUCS-005-96 (1996).
[55] G. Gupta, Introduction to data mining with case studies, PHI Learning Pvt. Ltd., 2014.
[56] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information processing & management 24 (5) (1988) 513–523.
[57] C. Soguero-Ruiz, K. Hindberg, I. Mora-Jiménez, J. L. Rojo-Álvarez, S. O. Skrøvseth, F. Godtliebsen, K. Mortensen, A. Revhaug, R.-O. Lindsetmo, K. M. Augestad, R. Jenssen, Predicting colorectal surgical complications using heterogeneous clinical data and kernel methods, Journal of Biomedical Informatics 61 (2016) 87–96.
[58] S. Robertson, Understanding inverse document frequency: on theoretical arguments for idf, Journal of documentation 60 (5) (2004) 503–520.
[59] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (85) (2008) 2579–2605.
[60] I. Fellows, R package 'wordcloud' (2012).
URL `https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf`
[61] H. Liu, T. Liu, J. Wu, D. Tao, Y. Fu, Spectral ensemble clustering, in: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2015, pp. 715–724.

[62] H. Wang, H. Shan, A. Banerjee, Bayesian cluster ensembles, Statistical Analysis and Data Mining 4 (1) (2011) 54–70.
[63] D. M. Witten, R. Tibshirani, A framework for feature selection in clustering, Journal of the American Statistical Association 105 (490) (2010) 713–726.
[64] H. Zhang, R. H. Zamar, A natural framework for sparse hierarchical clustering, in: arXiv preprint arXiv:1409.0745, 2014.
[65] K. Chaudhuri, S. Dasgupta, S. Kpotufe, U. von Luxburg, Consistent procedures for cluster tree estimation and pruning, IEEE Transactions on Information Theory 60 (12) (2014) 7900–7912.
[66] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.
[67] D. Yan, L. Huang, M. I. Jordan, Fast approximate spectral clustering, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 907–916.
[68] A. Vedaldi, S. Soatto, Quick shift and kernel methods for mode seeking, in: Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV, Springer Berlin Heidelberg, 2008, pp. 705–718.
[69] H.-S. Park, C.-H. Jun, A simple and fast algorithm for k-medoids clustering, Expert Systems with Applications 36 (2) (2009) 3336–3341.
[70] C. Meyer, S. Race, K. Valakuzhy, Determining the number of clusters via iterative consensus clustering, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 94–102.
[71] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

Appendix A. Clustering by evidence accumulation

This paper represents a contribution in the framework of clustering by mode seeking. Still, we have to acknowledge the perhaps most known method for ensemble or consensus clustering, the work of Fred and Jain [50] where k-means is used repeatedly with different initializations and random selections of the number of clusters to build a clustering ensemble. As in our method, single link and average link is used to calculate the consensus over the repeated clusterings.

Comparing with our proposed method, Fred and Jain's approach will in several cases give relatively similar results in practice, but due to the benefits of the kNN mode seeking algorithm it is in all cases much faster. To give a concrete example of this we run both k-means and kNN mode seeking on the MNIST data set, with varying number of clusters for k-means and varying number of nearest neighbors for our method[8]. The results are shown in Figure A.15 and the benefits of the kNN mode seeking algorithm becomes quite clear. K-means gets slower as the number of clusters increase, while kNN mode seeking stays close to constant speed regardless of the number of clusters. Finally, we compare our proposed algorithm with the k-means based algorithms of Fred and Jain on a selection of datasets from the UCI repository [71]. We report both clustering results in terms of adjusted rand index as well as the time taken for the clustering in Table A.12.

Reviewing the results shows us that the performance of our proposed algorithm is in most cases equal or better than both the single and average link version of the algorithm of Fred and Jain. In terms of speed the benefits of our algorithm are again obvious.

[8]We note that the number of clusters in kNN mode seeking is roughly inversely proportional to the number of nearest neighbors, larger neighborhoods yields lower number of clusters and vice versa

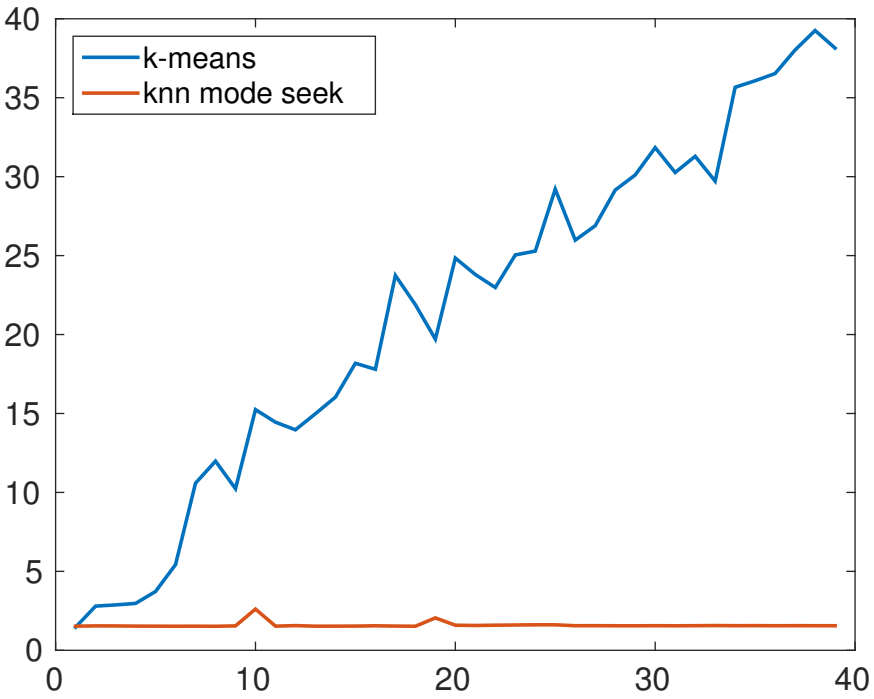


Figure A.15: Running times of kNN mode seeking vs k-means. The horizontal axis shows the number of clusters for k-means and the number of nearest neighbors for kNN mode seeking.

data set	EC_{kNN-MS}	EAC-SL	EAC-AL
cancer	.8070 (29s)	.0026 (114s)	.8018 (114s)
iris	.7592 (.8s)	.5659 (35.5s)	.5659 (35.9s)
wine	.3749 (1.3s)	.3007 (45.3s)	.3627 (50.6s)
crab	.5096 (1.4s)	- (82s)	.5631 (84.8s)

Table A.12: Clustering and running times results comparing EC_{kNN-MS} and the methods of Fred and Jain.

Biographies

Robert Jenssen (ansatte.uit.no/robert.jenssen) directs the Machine Learning @ UiT Lab: site.uit.no/ml at the Department of Physics and Technology, University of Tromsø (UiT) – The Arctic University of Norway. The group is partially funded by the Norwegian Research Council over FRIPRO grant 239844 on developing the Next Generation Learning Machines, focusing especially on information theoretic learning, kernel methods, and deep learning. Jenssen is Professor II at the Norwegian Computing Center in Oslo, and a senior researcher at the Norwegian Center on E-Health Research at the University Hospital of North Norway. He was a guest professor at the Technical University of Denmark, in the Cognitive Systems Section headed by Lars Kai Hansen in 2012/2013, a guest researcher at the Technical University of Berlin, in Klaus-Robert Müller's Machine Learning Group in 2008/2009, and a guest PhD student at the University of Florida, in Jose Principe's Computational NeuroEngineering Laboratory in 2002/2003 and March/April 2004. Jenssen

received Honorable Mention for the 2003 Pattern Recognition Journal Best Paper Award, the 2005 IEEE ICASSP Outstanding Student Paper Award, and the 2007 UiT Young Investigator Award. His paper "Kernel Entropy Component Analysis" was the Featured Paper of the May 2010 issue of IEEE Transactions on Pattern Analysis and Machine Intelligence, and his paper "Kernel Entropy Component Analysis for Remote Sensing Image Clustering" (with Gomez-Chova and Camps-Valls) was the Editor's Choice Paper of the March 2012 issue of the IEEE Geoscience and Remote Sensing Letters, and went on to win the 2013 IEEE Geoscience and Remote Sensing Society Letters Best Paper Award. Jenssen is a member of the IEEE Technical Committee on Machine Learning for Signal Processing, serves on the IAPR Governing Board, and is an Associate Editor of the journal Pattern Recognition. Jenssen is the general chair of the Scandinavian Conference on Image Analysis (SCIA) 2017.