Constrained Mean Shift Clustering

Maximilian Schier*† Christoph Reinders*† Bodo Rosenhahn†

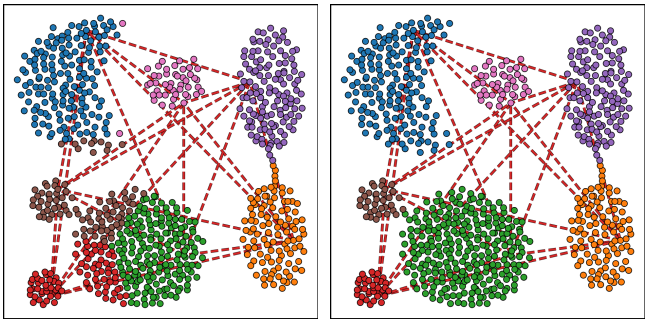
Abstract

In this paper, we present Constrained Mean Shift (CMS), a novel approach for mean shift clustering under sparse supervision using cannot-link constraints. The constraints provide a guidance in constrained clustering indicating that the respective pair should not be assigned to the same cluster. Our method introduces a density-based integration of the constraints to generate individual distributions of the sampling points per cluster. We also alleviate the (in general very sensitive) mean shift bandwidth parameter by proposing an adaptive bandwidth adjustment which is especially useful for clustering imbalanced data sets. Several experiments show that our approach achieves better performance compared to state-of-the-art methods both clustering synthetic data sets as well as clustering encoded features of real-world image data sets.

1 Introduction

Cluster analysis is the general task of grouping data samples based on the similarity of features. Such similarity is often determined through a distance metric, for example two data samples with real-valued features close in Euclidean distance are considered to be similar. Different approaches for cluster analysis are well known, among the most famous is k-Means [18], which aims to cluster data points into k clusters, such that the sum of Euclidean distances within each cluster is minimal. k-Means works by initializing k cluster centers randomly from the data points. Each data point is then assigned to the closest cluster center and the cluster center is updated as the mean of its assigned points. These two steps are repeated until convergence.

Mean shift [9] is another well-known clustering algorithm which has been widely used in applications like curve fitting, image segmentation, self-supervised feature learning, and road network detection [5,8,11,15]. The method uses the data points to estimate the density of the feature space. This is done for any point in feature space by determining the mean of all data points weighted by the kernel, effectively a window. Starting from each data point, mean shift repeatedly shifts the current point to the sample mean following the gradient of the density estimation. Points settling in the same local maxima, a mode, belong to the same cluster. Both k-Means and mean shift are unsupervised. Mean shift has well-known advantages compared to k-Means: the ability to cluster along nonlinear decision boundaries allowing complex cluster shapes and being less sensitive to cluster imbalance.



(a) PCKMeans [3] (b) CMS (ours)

Figure 1: Comparison of state-of-the-art constrained centroid-based clustering to our proposed method, constrained mean shift, on the *aggregation* data set [12] using the cannot-link constraints shown as red lines. Our novel approach to integrate cannot-link constraints into mean shift enables weak supervision, eliminates the common problem of correct bandwidth estimation, and outperforms state-of-the-art constrained clustering methods, especially on imbalanced data sets or for non-linearly separable clusters. (Best viewed in color)

In semi-supervised clustering, small amounts of binary constraints are provided as weak supervision [10]. Binary constraints are either must-link constraints, indicating that two data points should belong to the same cluster, or cannot-link constraints, expressing that two points should belong to different clusters. While some methods have been presented for k-Means that integrate constraints [3,25], binary constraints were not directly integrated into mean shift so far.

In this work, we present a novel constrained clustering method called *Constrained Mean Shift* (CMS), that combines the advantages of density-based mean shift clustering and weak supervision through binary

*Equal contribution

†Institute for Information Processing, Leibniz University Hannover. {schier, reinders, rosenhahn}@tnt.uni-hannover.de

constraints (see Fig. 1). Our approach integrates constraints by reducing sampling attraction via the kernel distance and generates an individual data distribution per cluster point for sampling. We also propose scaling the constraints to enable clustering on different scales and alleviate the common mean shift problem of finding a suitable bandwidth. Our contributions are:

- A novel constrained clustering method integrating binary cannot-link constraints into feature space mean shift clustering.
- Introduction of an adaptive bandwidth to enable accurate clustering on different scales and imbalanced data.
- The proposed approach outperforms other centroid-, density-, and eigenvalue-based clustering methods in feature space on synthetic data sets and real-world image embeddings.
- Our implementation of Constrained Mean Shift clustering is published.[1]

2 Related Work

Many cluster algorithms can be classified as centroid-based, density-based, hierarchical, and eigenvalue-based. Centroid-based algorithms aim to represent each cluster with a single prototype centroid and assign instances by distance to these centroids, such as k-Means [18]. Density-based approaches like mean shift [9], DBSCAN [6], or OPTICS [2] use a feature space density estimation to assign cohesive areas of high density to the same cluster. Hierarchical clustering approaches such as Ward clustering [27] build a hierarchy of clusters by using a linkage criterion based on a distance metric. Eigenvalue-based approaches like spectral clustering [26] use the eigenvalues of a similarity matrix of instances for clustering. As our approach extends mean shift, it falls in the category of density-based clustering.

Constrained Clustering One of the earliest works integrating binary cannot-link and must-link constraints into the k-Means algorithm was COP-k-Means [25], which uses the constraints as a hard prior by preventing assignment of instances to a cluster center which would violate a constraint. Since assignments are made sequentially with random order, an instance may have no assignable cluster without violation, in which case the algorithm fails. PCKMeans [3] improves stability by integrating constraint violation into the assignment cost function so that violations do not result in failure but only higher cost. Regular DBSCAN finds clusters by identifying strongly connected neighborhoods of points. C-DBSCAN [21] extends DBSCAN with binary constraints by breaking apart such clusters along cannot-link constraints and merging clusters with must-link constraints. Binary constraints have also been integrated into spectral clustering [26] by encoding soft constraints as a constraint matrix which is used as an auxiliary condition when solving the minimum graph cut for the regular spectral clustering problem.

Kernel & Deep Clustering Since clustering is largely dependent on the quality of features, different transformations were proposed to improve overall clustering results. Semi-supervised kernel clustering transforms the input space using a learned kernel into kernel space in a way that must-link instances are likely to be clustered alike and cannot-link instances differently. Kulis *et al.* [16] were among the first combining such a kernel with k-Means clustering improving results for nonlinearly separable clusters. For mean shift clustering Anand *et al.* [1] have proposed a similar method called SKMS, which learns a kernel using the binary constraints and performs unsupervised mean shift clustering in kernel space. Thus, the constraints are not directly used in the mean shift phase of the algorithm, but rather used to shape an easy to cluster kernel space.

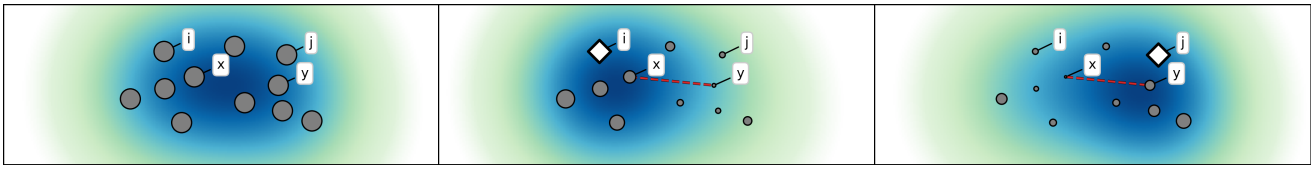
More recently, cluster algorithms have been combined with deep learning. In Deep Embedding Clustering (DEC) [29] an autoencoder network is trained on an image data set, then its embeddings are hardened using a k-Means like loss function. Semi-supervision through binary constraints as a soft prior was integrated into this procedure by Ren *et al.* [20] by adding a constraint violation loss.

3 Mean Shift

We briefly summarize regular mean shift [9] in this section. Mean shift is a density-based clustering algorithm estimating the d-dimensional feature space density using a kernel K and shifting cluster centers by ascending the gradient of this estimation. Valid kernel functions $K : \mathbb{R}^d \to \mathbb{R}$ must have a profile $k : [0, \infty] \to [0, 1]$ such that $K(\mathbf{x}) = k(\|\mathbf{x}\|_2^2)$ and k is monotonically decreasing, piecewise continous, and its integral finite. Given n data points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, the sampling points $\mathbf{S}$ and cluster centers $\mathbf{T}^{(0)}$ are commonly initialized with the data points, i.e. $\mathbf{s}_i = \mathbf{x}_i$, $\mathbf{t}_i^{(0)} = \mathbf{x}_i$. Using a specified bandwidth h, intuitively the scale of the kernel, mean shift estimates the new sample mean of any point $\mathbf{x}$ in feature space as:

$$m(\mathbf{x}) = \frac{\sum_{j=1}^{n} k\left(\left\|\frac{\mathbf{s}_j - \mathbf{x}}{h}\right\|^2\right) \cdot w(\cdot) \cdot \mathbf{s}_j}{\sum_{j=1}^{n} k\left(\left\|\frac{\mathbf{s}_j - \mathbf{x}}{h}\right\|^2\right) \cdot w(\cdot)}, \tag{3.1}$$

[1] https://github.com/m-schier/cms



(a) KDE of regular mean shift (b) KDE of CMS when shifting i (c) KDE of CMS when shifting j

Figure 2: Influence of constraints on kernel density estimation (KDE) in (constrained) mean shift. Grey circles mark the cluster centers where the size indicates the weighting $w(\cdot)$. Sampling points are identical to the cluster centers. For CMS, the current cluster center is marked as a white diamond. CMS estimates kernel density individually for each cluster center, reducing the weight of sampling points close to a cannot-link constraint.

where $w(\cdot)$ is a function for weighting each sample $\mathbf{s}_j$. In the common case it is simply a constant $w = 1$. We will later use $w(\cdot)$ for our proposed integration of cannot-link constraints. The clustering is performed by shifting all cluster centers to their sampling mean in each iteration u: $\mathbf{T}^{(u)} = \left\{m\left(\mathbf{t}_1^{(u-1)}\right), \ldots, m\left(\mathbf{t}_n^{(u-1)}\right)\right\}$. This shifting is repeated until the maximum number of iterations $u_{\max}$ or convergence. The outlined approach of mean shift is often referred to as *nonblurring* [5] as the sampling points $\mathbf{S}$ remain stationary. The approach proposed by Fukunaga and Hostetler [9] uses the previous cluster centers $\mathbf{T}^{(u-1)}$ as sampling points $\mathbf{S}$ in each iteration u, except for the first iteration, where the data points $\mathbf{X}$ are used. Since the sampling points are fully blurred, this version is commonly referred to as *blurring* mean shift. Our proposed approach for constraints works on both methods and only requires sampling points and cluster centers of the current iteration, therefore the time indices are omitted for readability in the following.

4 Constrained Mean Shift

Our proposed method enables semi-supervised density-based clustering with mean shift. In this section we describe our main contributions: integrating binary cannot-link constraints into the sample mean, constraint scaling, and adaptive bandwidth.

4.1 Integrating Constraints Given n data points in d-dimensional space $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, let C be the binary relation of cannot-link constraints, i.e. $(i, j) \in C$ indicates that the i-th and j-th data point must not be assigned to the same cluster. Since both the sampling points $\mathbf{S}$ and cluster centers $\mathbf{T}$ are initialized with the data points $\mathbf{X}$, the cannot-link constraints C can relate between cluster centers directly, but also between cluster centers and sampling points. Therefore, $(i, j) \in C$ indicates $\mathbf{t}_i$ and $\mathbf{t}_j$ should not be close and that $\mathbf{t}_i$ should not sample from $\mathbf{s}_j$ and vice versa. In Eq. (3.1), we highlighted the use of a weighting function $w(\cdot)$ during the mean shift. We introduce the integration of constraints by designing a weighting function $w(\mathbf{t}_i, \mathbf{s}_j)$ returning the sampling weight of $\mathbf{s}_j$ when sampling for the cluster center $\mathbf{t}_i$. A naive approach would be not sampling from $\mathbf{s}_j$ if a constraint exists: $w(\mathbf{t}_i, \mathbf{s}_j) = [(i, j) \notin C]$, where $[\cdot]$ is the Iverson bracket operator. However, this would not prevent sampling in the vicinity of $\mathbf{s}_j$, in which case a high density region close to $\mathbf{s}_j$ would still be sampled from and heavily influence $\mathbf{t}_i$, despite being very similar in features to a sampling point which $\mathbf{t}_i$ should not attract to.

Therefore we present a distance-based integration of constraints to determine individual attractions between cluster centers and sampling points. Given a constraint $(x, y) \in C$, with increasing closeness of cluster centers $\mathbf{t}_i$ to $\mathbf{t}_x$ and $\mathbf{t}_j$ to $\mathbf{t}_y$ the sampling weight of $\mathbf{s}_j$ should be reduced. This idea is illustrated in Fig. 2. For regular mean shift, all samples are weighted equally so that the kernel density estimation is influenced equally by all sampling points and is identical for all cluster centers. When introducing a constraint between the points labeled x and y, cluster centers on the right side of the constraint should attract less to sampling points on the left side and vice versa. Since this constraint-driven reduction should become 0 (and therefore the weighting $w(\cdot)$ becomes 1) as soon as either the cluster center or the sampling point are no longer close to the constraint as determined by the kernel, we can multiply both kernel responses and subtract the result from 1. Thus, we propose the constraint-based weighting function $R(\mathbf{t}_i, \mathbf{s}_j, (x, y))$ to calculate the multiplicative weight reduction caused by the constraint $(x, y) \in C$ when sampling $\mathbf{s}_j$ for cluster center $\mathbf{t}_i$:

(4.2)
$$R(\mathbf{t}_i, \mathbf{s}_j, (x, y)) = 1 - k\left(\left\|\frac{\mathbf{t}_x - \mathbf{t}_i}{h_c}\right\|^2\right) k\left(\left\|\frac{\mathbf{t}_y - \mathbf{t}_j}{h_c}\right\|^2\right),$$

where h_c is the bandwidth that should be used for

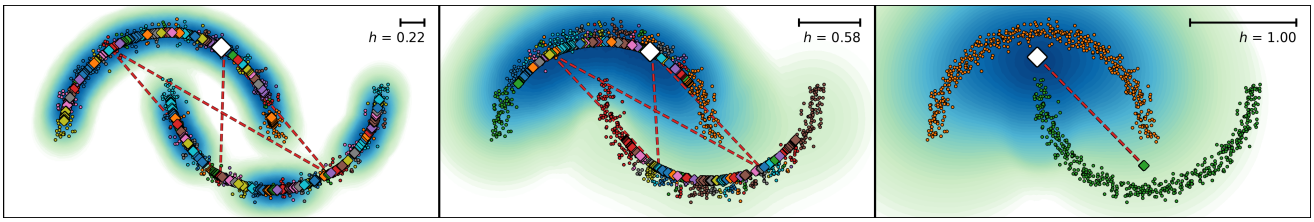


Figure 3: CMS integrates cannot-link constraints (dashed red lines) by reducing the kernel density estimation (KDE) on opposite ends of constraints. Background contour shows KDE for the cluster center (diamond) highlighted in white. As the bandwidth h is increased with the iterations through our proposed adaptive bandwidth (left to right), the local structures converge to a stable global clustering. KDE in CMS completely ignores all points of the opposite cluster in this stable state through our proposed weighting.

the constraint. We have found it beneficial to use a different bandwidth than the global bandwidth h and will provide details on individual bandwidths per constraint in the next section. Note that we are measuring the distance of cluster centers on both the side of the cluster center ($\mathbf{t}_i$ to $\mathbf{t}_x$) and the side of the sampling point ($\mathbf{t}_j$ to $\mathbf{t}_y$). Since the cluster centers are updated during iterations of mean shift while the sampling points remain stationary in *nonblurring* mean shift, the cluster centers better reflect the density of the data. Thus, when calculating constraint weights on cluster centers, the constraints move with the evolving estimation of the data density. This is apparent when viewing Fig. 3 where the constraints anchored at the cluster centers are moving inwards through iterations, thus at the end all cluster centers are very close to one of the two cluster centers on either moon with a constraint.

As $R(\cdot)$ is 1, the multiplicative identity, if a constraint has no reduction on sampling, the reduction through all constraints can be combined by multiplication. Thus, the weight function $w(\mathbf{t}_i, \mathbf{s}_j)$ is defined as:

$$w(\mathbf{t}_i, \mathbf{s}_j) = \prod_{(x,y)\in C} R(\mathbf{t}_i, \mathbf{s}_j, (x, y)). \tag{4.3}$$

Because the relation of C is symmetric, all constraints are covered for both points.

4.2 Constraint Scaling In Eq. (4.2), we defined the constraint bandwidth h_c, the bandwidth of the kernel when measuring proximity to a constraint. If h_c would always be equal to the global bandwidth h and a given constraint would be of very small scale compared to h, then any point surrounding the constraint would be nearly equally close to both ends of the constraint as measured by the kernel. Such a constraint would provide no useful information and only hinder convergence, reducing all sampling weights in its proximity equally. However, the weak supervision of the constraint provides additional information regarding the local scale of the data distribution. To enable clustering on data sets with largely varying constraint scales, we introduce an individual bandwidth per constraint. Intuitively, the bandwidth used to determine proximity to a constraint should scale with the length of the constraint. Thus, given two cluster centers $\mathbf{t}_x$ and $\mathbf{t}_y$ connected through a constraint, we propose scaling by $\lambda \cdot \|\mathbf{t}_x - \mathbf{t}_y\|_2$ with λ as a constant factor. Since the bandwidth per constraint should never exceed the global bandwidth h and not become zero, we clip the bandwidth to the valid range:

$$h_c((x, y)) = \max\{\epsilon, \min\{h, \lambda \cdot \|t_x - t_y\|_2\}\} \tag{4.4}$$

with ϵ as some reasonably small minimum value depending on the computing platform. We study the influence of λ in our experiments and ablation studies.

4.3 Adaptive Bandwidth The regular mean shift uses a constant global bandwidth h for all iterations of the process. This bandwidth influences the number of clusters by defining the number of modes of the kernel density estimation. With low h many undesired clusters are formed, in the trivial case of h being close to 0 every sampling point has its own mode. If h is too large, modes which should be distinct will collapse. Thus, a key problem of mean shift is estimating the parameter h correctly, such that it is as large as possible without causing mode collapse.

In CMS, our introduced constraints prevent mode collapse, therefore we enable starting with a low bandwidth and increasing the bandwidth until it is large enough for all cluster centers to converge on modes only separated by constraints. This approach is illustrated in Fig. 3. By starting with a low bandwidth, the first iterations shift points towards local modes, which is beneficial if constraints do not cover the entire feature space. In the final step, a stable clustering is reached, as for arbitrarily high bandwidths our introduced constraints

Method \ Data set	moons		aggregation		jain		s4	
	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
Mean shift	0.878	0.799	0.777	0.838	0.464	0.495	0.112	0.346
C-DBSCAN	**1.000**	**1.000**	0.969	0.968	0.958	0.937	0.430	0.653
Constrained Spectral	0.207	0.194	0.774	0.766	0.995	0.989	0.071	0.243
MPCKMeans	0.627	0.542	0.729	0.832	0.868	0.785	0.597	0.717
PCKMeans	0.872	0.793	0.751	0.854	0.921	0.856	**0.627**	**0.732**
CMS (**ours**)	0.996	0.996	**0.987**	**0.983**	**1.000**	**1.000**	0.618	0.728

Table 1: Comparison of the performance on commonly used synthetic "toy" data sets. Best results are highlighted in bold, second best are underlined. Our proposed method CMS performs best or close to the best on all tested data sets. CMS always improves upon the unsupervised baseline, mean shift.

prevent further collapse. We propose interpolating linearly between an initial low bandwidth $h_{\min}$ and final large bandwidth $h_{\max}$ from iteration 0 to the final iteration $u_{\max}$. The bandwidths $h_{\min}$ and $h_{\max}$ can be determined by taking the smallest nonzero and largest Euclidean distance between data points, respectively.

5 Experiments

The performance of our proposed constrained mean shift (CMS) algorithm is evaluated on synthetic data sets and encoded features of real-world image data sets. We compare CMS with regular mean shift and several state-of-the-art constrained clustering algorithms operating directly in feature space. The following methods are used for comparison:

Mean shift [9] Unconstrained regular mean shift using the same kernel as CMS (included as baseline)

PCKMeans [3] k-Means [18] clustering with the addition of binary must-link and cannot-link constraints as soft prior

MPCKMeans [4] Extension of PCKMeans which also estimates a suitable metric in the form of a Mahalanobis distance

C-DBSCAN [21] Constrained version of regular Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6]

Constrained Spectral [26] A constrained spectral clustering approach introduced by Wang *et al.*

5.1 Evaluation We follow Ren *et al.* [20] and use Adjusted Rand Index (ARI) [13] and Normalized Mutual Information (NMI) [23] for performance analysis. The ARI is a modification of the Rand Index (RI) [19] comparing the number of pairs of instances correctly assigned to the same class and correctly assigned to different classes divided by the total number of pairs. This measure is adjusted using the expectation of the RI such that the ARI is in range $[-1, 1]$, where 1 corresponds to a perfect assignment, 0 is random, and lower is worse than random. The NMI is a comparison metric of the mutual information between a predicted class assignment and ground truth classes normalized by the entropy of both distributions. NMI lies in the range $[0, 1]$ where 1 corresponds to a perfect assignment. Both these metrics better reflect the quality of the clustering result than clustering accuracy, especially on imbalanced data.

5.2 Experimental Setup In order to generate binary constraints, the ground truth class labels are used. For a desired number of constraints n_c, pairs of data points are sampled and added as must-link constraints if they belong to the same class, otherwise as cannot-link constraints, until a total of n_c constraints is reached. During sampling, we ensure that at least one cannot-link constraint exists between all pairs of classes. This allows a fair comparison between cluster methods which are given the correct number of clusters as input and those determining cluster count from data. Similar to other authors [3, 25] we calculate the transitive closure of the constraints to explicitly add knowledge logically implied by the given constraints. For preprocessing min-max normalization is performed on all features independently before clustering. All experiments are repeated 10 times. For each repetition, the used subset of the data set and the constraints are randomly sampled.

Unless noted otherwise, we set the hyperparameters of CMS constraint scale $\lambda = 0.5$, number of iterations $u_{\max} = 80$, and use a truncated radial basis function (RBF) kernel $K_{\text{trunc}}(\cdot)$ with truncation $c = 0.2$:

$$K_{\text{trunc}}(\mathbf{x}) = \begin{cases} \exp\left(-\|\mathbf{x}\|_2^2\right) & \text{if } \exp\left(-\|\mathbf{x}\|_2^2\right) > c \\ 0 & \text{otherwise.} \end{cases}$$

For other algorithms, we determine a set of hyperparameters with best average ARI over all data sets through grid search, as tuning on a data set level would be unrealistic given the lack of ground truth data in real appli-

Method \ Data set	MNIST		Fashion-MNIST		GTSRB	
	ARI	NMI	ARI	NMI	ARI	NMI
Mean shift	0.336	0.567	0.387	0.572	0.331	0.572
C-DBSCAN	0.271	0.561	0.249	0.490	0.263	0.532
Constrained Spectral	0.003	0.043	0.008	0.057	0.007	0.056
MPCKMeans	0.605	0.676	0.418	0.545	0.445	0.549
PCKMeans	0.650	0.708	0.407	0.542	0.472	0.573
CMS (**ours**)	**0.754**	**0.783**	**0.452**	**0.611**	**0.553**	**0.654**

Table 2: Performance of constrained mean shift and compared methods on image embeddings of real-world data sets measured by Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). Best results are highlighted in bold.

cations. For regular mean shift, we estimate the bandwidth h to be the 15th percentile of all pair-wise distances. For C-DBSCAN we set minpts $= 2$ and $\epsilon = 0.1$ for synthetic and $\epsilon = 0.31$ for image data sets as no setting is well suited for both classes of data.

For our evaluation on image data we do not use the raw pixel features, since previous work has shown that feature space clustering approaches do not scale well to high dimensional image data [29]. Therefore, we generate features using an established deep neural network autoencoder architecture [20, 29]. The autoencoder is a multi-layer perceptron. Encoder dimensions are d-500-500-2000-10, where d is the flattened input shape and the latent space dimension is 10. The decoder has a mirrored architecture. All layers have ReLU activations, except for the layers predicting the latent space and the reconstructed output, which have no activation. One instance of the autoencoder is trained per data set using a stacked denoising approach [24] as described by Xie *et al.* [29] with the reconstruction loss, constraints or class labels are not used for supervision. The embeddings of all images per data set are predicted and serve as features for clustering of the images.

5.3 Synthetic Data Sets We compare CMS with other state-of-the-art clustering algorithms on well-known toy examples: the regular *moons* data set already shown in Fig. 3, a version with one moon of much lower density, *jain* [14], a data set of Gaussian clusters connected through single links, *aggregation* [12] as shown in Fig. 1, and a data set of overlapping Gaussians with axis-independent scaling, *s4* [7]. For comparison, we use the full data set as made available by the respective authors, except for *moons*, which is dynamically generated with 500 instances. Per data set, we sample a number of constraints equal to the number of instances.

The results are shown in Tab. 1. CMS achieves near-perfect score on the *moons* data set and perfect score on *jain*. These results highlight that CMS improves on the strength of mean shift in clustering along nonlinear decision boundaries. Regular mean shift achieves lower performance as the tips of the moons are likely to be incorrectly clustered without constraints. *aggregation* has clusters of varying extent and quantity. k-Means-based methods do not perform well in such cases. CMS, however, achieves very good performance by determining suitable bandwidths locally and globally through our proposed bandwidth scaling. On *s4*, which is a harder example due to overlapping clusters, CMS performs slightly worse than PCKMeans, but better than all other tested methods. Overall, CMS reliably performs best or close to best and always improves upon the results obtained by regular mean shift.

5.4 Image Data Sets We compare semi-supervised clustering performance on real-world image data sets. The greyscale data sets MNIST [17] consisting of handwritten digits and Fashion-MNIST [28] consisting of clothing and fashion products are used. Both data sets contain ten mostly balanced classes and have an image resolution of 28×28 pixels. We also use the German Traffic Sign Recognition Benchmark (GTSRB) [22] consisting of colored images of traffic signs. Since GTSRB includes 43 classes, we use a subset of 10 classes shown in Fig. 5. Furthermore, we scale all GTSRB images to 32×32 pixels and normalize brightness using histogram stretching.

First, we compare the performance of all clustering algorithms on the generated ten-dimensional embeddings in terms of ARI and NMI by randomly sampling a subset of 2,000 image embeddings and 2,000 constraints. The results are shown in Tab. 2. CMS achieves an ARI of 0.754 on MNIST, which is 0.418 better than unconstrained mean shift with an ARI of 0.336, and 0.104 better than the next best constrained clustering method. On GTSRB, CMS reaches an ARI of 0.553 compared to 0.331 of unconstrained mean shift and also outperforms the next best constrained clustering algorithm by

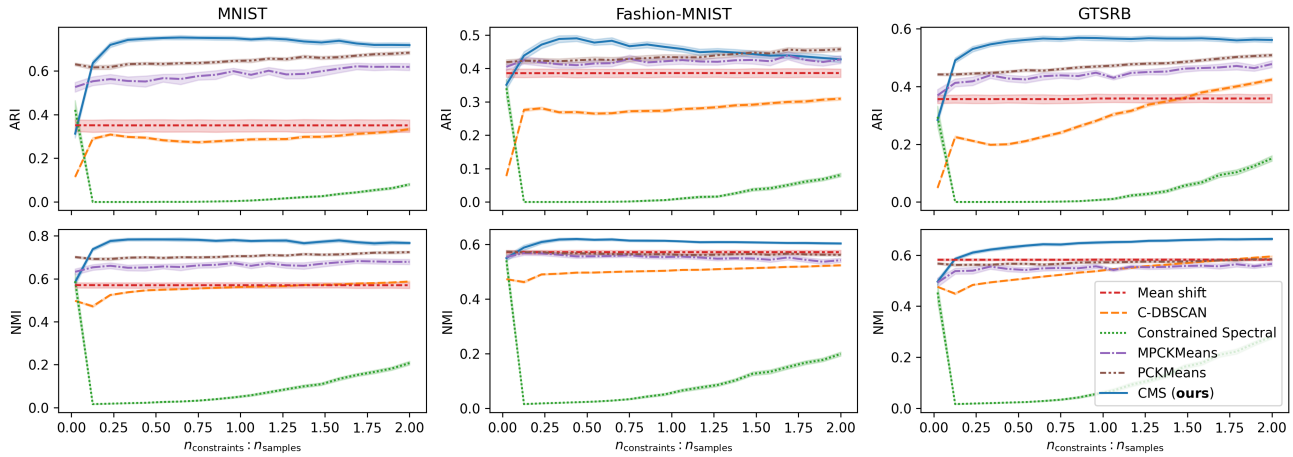


Figure 4: Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) of several constrained cluster methods on MNIST [17], Fashion-MNIST [28], and GTSRB [22] embeddings for varying numbers of constraints.

Figure 5: German Traffic Sign Recognition Benchmark (GTSRB) classes used in the experiments. Top row: pictograms of the classes, below: two random examples from the data set after preprocessing.

a margin of 0.081 ARI. Overall, CMS achieves the best performance in terms of both metrics on all data sets.

Next, the impact of the amount of constraints is evaluated. Again 2,000 embeddings are sampled while the number of generated constraints is varied from the minimal amount of 45 up to 4,000. The results are shown in Fig. 4. On MNIST and GTSRB, CMS achieves the best performance starting from a constraint ratio of 0.1, which corresponds to 200 constraints. For less constraints k-Means-based methods perform better. On Fashion-MNIST, CMS outperforms all other methods from a range of 200 to 3,000 constraints in ARI, for a larger number of constraints PCKMeans performs slightly better. Fashion-MNIST is difficult to cluster as many of the classes are highly overlapping in feature space. Adding many constraints causes individual clusters to be formed by CMS in these overlapping regions which are not merged. Thus, the number of clusters increases, negatively impacting ARI. For this reason, ARI decreases starting from a constraint ratio of 0.4. In terms of NMI, however, CMS performs best starting from very few constraints.

Finally, we evaluate the performance on imbalanced data sets. Similar to Xie *et al.* [29], we sample with a retention rate r, where samples of the first class are given a weight of r during random selection, samples of the last class a weight of 1, and all classes in between have a linearly interpolated sampling weight. The order of the classes is also randomized for each run of the experiment. We always sample 2,000 instances and 1,000 constraints. The results on the MNIST data set are shown in Tab. 3. CMS achieves the best performance for all amounts of imbalance tested, showing that the high performance achieved on balanced data can be maintained for very unbalanced clusters as well.

5.5 Ablation Studies To study the importance of our main contributions, we individually disable the cannot-link sampling weighting (Sec. 4.1), the constraint scaling (Sec. 4.2), and the adaptive bandwidth (Sec. 4.3) while clustering GTSRB embeddings. With no adaptive bandwidth we determine an optimal constant global bandwidth h by grid search. Without constraint scaling, all constraints are scaled by the global bandwidth h. The results are shown in Tab. 4. Without constraints (M1), CMS works like a regular mean shift. Using an adaptive bandwidth without constraints causes a mode collapse (M2). Integrating constraints without scaling (M3 and M4) does not exceed the baseline, as constraints start blocking clustering. With a constant bandwidth, constraint scaling shows improved performance over the previous results (M5). Combining all contributions (M6), the best performance is achieved. Thus, each of our main contributions plays an important role in the good performance of CMS.

Furthermore, we evaluate the performance impact of different kernels as well as both *blurring* and *nonblur-*

Method \ r	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Mean shift	0.370	0.361	0.439	0.374	0.354	0.304	0.373	0.351	0.344	0.362
C-DBSCAN	0.263	0.293	0.323	0.299	0.266	0.312	0.294	0.303	0.285	0.305
Constrained Spectral	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MPCKMeans	0.542	0.559	0.556	0.492	0.575	0.510	0.580	0.560	0.585	0.569
PCKMeans	0.591	0.584	0.609	0.585	0.625	0.603	0.649	0.606	0.637	0.638
CMS (**ours**)	**0.755**	**0.764**	**0.777**	**0.734**	**0.752**	**0.732**	**0.770**	**0.756**	**0.736**	**0.765**

Table 3: Adjusted Rand Index (ARI) on imbalanced subsample of MNIST [17] embeddings. Imbalance increases with lower retention rate r. Best results are highlighted in bold. CMS performs best on all imbalances tested and is not adversely affected.

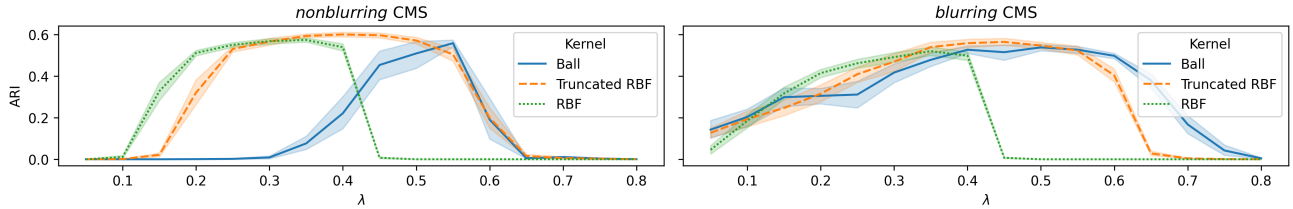


Figure 6: Adjusted Rand Index (ARI) of CMS over a range of constraint scales λ on GTSRB [22] embeddings for different mean shift kernels as well as *nonblurring* and *blurring* mean shift. CMS generalizes well to other kernels as well as other modes, performing comparably well for suitable values of λ.

	CL	CS	AB	ARI	NMI
M1				0.405	0.596
M2			✓	0.000	0.000
M3	✓			0.346	0.541
M4	✓		✓	0.000	0.447
M5	✓	✓		0.529	0.645
M6	✓	✓	✓	**0.574**	**0.677**

Table 4: Impact of our individual contributions - integrating cannot-link constraints (CL, Sec. 4.1), constraint scaling (CS, Sec. 4.2), adaptive bandwidth (AB, Sec. 4.3) - on the performance of CMS on GTSRB [22] embeddings. Best results are highlighted in bold.

ring mode on CMS to study its adaptability. Two new kernels are used, the ball kernel with profile $k_{\text{ball}}(x) = [x < 1]$ and the radial basis function kernel with profile $k_{\text{RBF}}(x) = \exp(-x)$. For all kernels and modes, the performance for λ values is shown in Fig. 6. CMS performs best using a truncated RBF kernel, although both other kernels are only slightly worse if a suitable λ value is selected for both the *blurring* and *nonblurring* variant. Thus, CMS generalizes well to different common mean shift kernels. Furthermore, the range of nearly optimal values of λ is quite large, therefore CMS is not very sensitive to this hyper-parameter.

6 Conclusion

In this paper, we presented the integration of cannot-link constraints into mean shift to combine density-based clustering and weak supervision. Our novel approach reduces the sampling weight depending on the proximity of current cluster center and sampling point to a constraint as estimated by the kernel. Furthermore, we introduced an additional constraint scaling to enable clustering on different scales, without small constraints preventing shifting on a globally large scale. Using an adaptive bandwidth, we integrate the local feature density while also settling into a globally stable clustering. We evaluated the performance of our proposed method, CMS, on synthetic data sets and features of real-world image data sets, obtained by an autoencoder. CMS achieves the best performance or very close to best performance on all synthetic data sets and performs much better than other state-of-the-art methods on image embeddings of MNIST and GTSRB. On highly imbalanced data sets, CMS also outperforms all other tested methods. We showed that CMS is stable in performance regarding changes to its single major hyper-parameter and generalizes well to different mean shift kernels.

Future work on CMS might focus on better integrating must-link constraints, which are currently only used for a transitive closure of the constraint preprocessing, and treating constraints as a soft prior.

7 Acknowledgements

This work was supported by the Federal Ministry of Education and Research (BMBF) Germany under the project LeibnizKILabor (grant no. 01DD20003), the Center for Digital Innovations (ZDIN), and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122).

References

[1] S. Anand, S. Mittal, O. Tuzel, and P. Meer, *Semi-supervised kernel mean shift clustering*, IEEE transactions on pattern analysis and machine intelligence, 36 (2013), pp. 1201–1215.

[2] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, *Optics: Ordering points to identify the clustering structure*, ACM Sigmod record, 28 (1999).

[3] S. Basu, A. Banerjee, and R. J. Mooney, *Active semi-supervision for pairwise constrained clustering*, in Proceedings of the 2004 SIAM international conference on data mining, SIAM, 2004, pp. 333–344.

[4] M. Bilenko, S. Basu, and R. J. Mooney, *Integrating constraints and metric learning in semi-supervised clustering*, in ICML, 2004, p. 11.

[5] Y. Cheng, *Mean shift, mode seeking, and clustering*, IEEE transactions on pattern analysis and machine intelligence, 17 (1995), pp. 790–799.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., *A density-based algorithm for discovering clusters in large spatial databases with noise*, in KDD, vol. 96, 1996, pp. 226–231.

[7] P. Fränti and O. Virmajoki, *Iterative shrinking method for clustering problems*, Pattern Recognition, 39 (2006), pp. 761–775.

[8] Z. Fu, L. Fan, Y. Sun, and Z. Tian, *Density adaptive approach for generating road network from gps trajectories*, IEEE Access, 8 (2020), pp. 51388–51399.

[9] K. Fukunaga and L. Hostetler, *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Transactions on information theory, 21 (1975), pp. 32–40.

[10] P. Gançarski, B. C. Thi-Bich-Hanh Dao, G. Forestier, and T. Lampert, *Constrained clustering: Current and new trends*, A Guided Tour of Artificial Intelligence Research, (2020), pp. 447–484.

[11] Y. A. Ghassabeh and F. Rudzicz, *Modified subspace constrained mean shift algorithm*, Journal of Classification, 38 (2021), pp. 27–43.

[12] A. Gionis, H. Mannila, and P. Tsaparas, *Clustering aggregation*, 21st International Conference on Data Engineering, (2005), pp. 341–352.

[13] L. Hubert and P. Arabie, *Comparing partitions*, Journal of classification, 2 (1985), pp. 193–218.

[14] A. K. Jain and M. H. Law, *Data clustering: A user's dilemma*, in International conference on pattern recognition and machine intelligence, Springer, 2005, pp. 1–10.

[15] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash, *Mean shift for self-supervised learning*, in Proceedings of the ICCV, 2021, pp. 10326–10335.

[16] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, *Semi-supervised graph clustering: a kernel approach*, Machine learning, 74 (2009), pp. 1–22.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[18] J. MacQueen et al., *Some methods for classification and analysis of multivariate observations*, in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, Oakland, CA, USA, 1967, pp. 281–297.

[19] W. M. Rand, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical association, 66 (1971), pp. 846–850.

[20] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. Hoi, and Z. Xu, *Semi-supervised deep embedded clustering*, Neurocomputing, 325 (2019), pp. 121–130.

[21] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, *C-dbscan: Density-based clustering with constraints*, in International workshop on rough sets, fuzzy sets, data mining, and granular-soft computing, Springer, 2007.

[22] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, *The german traffic sign recognition benchmark: a multi-class classification competition*, in The 2011 international joint conference on neural networks, IEEE, 2011, pp. 1453–1460.

[23] A. Strehl and J. Ghosh, *Cluster ensembles—a knowledge reuse framework for combining multiple partitions*, JMLR, 3 (2002), pp. 583–617.

[24] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.*, JMLR, 11 (2010).

[25] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., *Constrained k-means clustering with background knowledge*, in ICML, vol. 1, 2001, pp. 577–584.

[26] X. Wang, B. Qian, and I. Davidson, *On constrained spectral clustering and its applications*, Data Mining and Knowledge Discovery, 28 (2014), pp. 1–30.

[27] J. H. Ward Jr, *Hierarchical grouping to optimize an objective function*, Journal of the American statistical association, 58 (1963), pp. 236–244.

[28] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*, arXiv preprint arXiv:1708.07747, (2017).

[29] J. Xie, R. Girshick, and A. Farhadi, *Unsupervised deep embedding for clustering analysis*, in ICML, 2016, pp. 478–487.