Spectral Clustering using *PCKID* – A Probabilistic Cluster Kernel for Incomplete Data

Sigurd Løkse[1⋆], Filippo M. Bianchi[1], Arnt-Børre Salberg[2], and Robert Jenssen[1,2]

[1] Machine Learning Group[⋆⋆], UiT – The Arctic University of Norway
[2] Norwegian Computing Center

Abstract. In this paper, we propose *PCKID*, a novel, robust, kernel function for spectral clustering, specifically designed to handle incomplete data. By combining posterior distributions of Gaussian Mixture Models for incomplete data on different scales, we are able to learn a kernel for incomplete data that does not depend on any critical hyperparameters, unlike the commonly used RBF kernel. To evaluate our method, we perform experiments on two real datasets. *PCKID* outperforms the baseline methods for all fractions of missing values and in some cases outperforms the baseline methods with up to 25 percentage points.

Keywords: Missing data, robustness, kernel methods, spectral clustering

1 Introduction

Clustering is of utmost importance in the field of machine learning, with a huge literature and many practical applications [7]. Over the past decades, a huge variety of methods have been proposed. These range from simple linear methods like k-means [21], to more recent advanced methods, like spectral clustering [4, 14, 15, 22, 23]. Spectral clustering is a family of highly performing clustering algorithms, currently considered state of the art. In spectral clustering, the eigenvectors and eigenvalues (spectrum) of some similarity matrix are exploited to generate a beneficial representation of the data, such that a simple method like k-means could be utilized to generate a partitioning, even with non-linearly separable data.

Analyzing incomplete datasets (with missing features) is a big challenge within clustering methods and data analysis in general, since encountering incomplete data is common in real applications. For instance, an entry in the dataset may not be recorded if a sensor is failing or a field in a questionnaire is left unanswered. Both supervised and unsupervised methods have been proposed to deal with incomplete data. In the supervised setting, we have e.g. a max–margin framework, where geometric interpretations of the margin is used

⋆ sigurd.lokse@uit.no
⋆⋆ http://site.uit.no/ml

to account for missing data [1], an approach based on training one SVM per missingness pattern [17] and the "best" Bayesian classifier [12] approach. In the unsupervised setting, there are mixture model formulations accounting for missing features, including both non–Bayesian approaches [5, 10] and Bayesian approaches [11]. In general, a common approach is to apply imputation techniques [3] to estimate the missing values and then proceeding with the analysis on the imputed, complete, data set. None of these approaches come without challenges since the best choice of imputation technique is often very dependent on the data, and moreover difficult to evaluate.

In this paper, we propose as a new approach to integrate in a synergistic manner recent advances in spectral clustering and kernel methods with existing probabilistic methods for dealing with incomplete data. In particular, we exploit the Probabilistic Cluster Kernel (PCK) framework [6], which combines posterior distributions of Gaussian Mixture Models (GMMs) on different scales to learn a robust kernel function, capturing similarities on both a global and local scale. This kernel function is robust with regards to hyperparameter choices, since instead of assuming some structure in the data, the ensemble of GMMs adapt to the data manifold. We hyptothesize that by integrating GMMs specifically designed to handle incomplete data [10] into the PCK framework for spectral clustering, we will be able to cluster incomplete data sets in a more robust manner compared to existing approaches. The proposed approach for building the kernel matrix to be used for spectral clustering in our framework, is denoted the *Probabilistic Cluster Kernel for Incomplete Data (PCKID)*.

2 Background theory

2.1 Missing data mechanisms

Let $\mathbf{x} = \{x_i\}$ denote a data vector and let $\mathbf{x}^{\mathrm{o}}$ and $\mathbf{x}^{\mathrm{m}}$ denote the observed- and missing features of $\mathbf{x}$. Define $\mathbf{r} = \{r_i\}$, where $r_i = 1$ if $x_i \in \mathbf{x}^{\mathrm{m}}$ and zero otherwise to be the *missing indicator* for $\mathbf{x}$. In order to train a model that accounts for values in the dataset that are not observed, one has to rely on assumptions that describe how missing data occurs. In this section, we describe the three main missing data mechanisms that characterize the structure of $\mathbf{r}$ [17].

Missing completely at random (MCAR) Features are said to be *missing completely at random* (MCAR) if the features are missing independently from both the observed values $\mathbf{x}^{\mathrm{o}}$ and the missing values $\mathbf{x}^{\mathrm{m}}$. That is,

$$P(\mathbf{R}|\mathbf{X}) = P(\mathbf{R}).$$

This is the missingness assumption on the data that leads to the simplest analysis. However, this assumption is rarely satisfied in practice.

Missing at random (MAR) If the *features* are missing independently of their *values*, the features are said to be *missing at random* (MAR). Then the missingness of the features are only dependent of the *observed* values, such that

$$P(\mathbf{R}|\mathbf{X}) = P\left(\mathbf{R}|\mathbf{X}^{\mathrm{o}}\right).$$

This missing data mechanism is often assumed when working with missing data, since many real world missing data are generated by this mechanism. For instance, a blood test of a patient might be missing if it is only taken given some other test (observed value) exceeds a certain value.

Not missing at random (NMAR) If the missingness of a feature is dependent on their values, it is said to be not missing at random (NMAR), that is

$$P(\mathbf{R}|\mathbf{X}) = P\left(\mathbf{R}|\mathbf{X}^{\mathrm{m}}\right).$$

For instance, NMAR occurs when a sensor measurement is discarded because it goes beyond the maximum value that the sensor can handle.

2.2 Gaussian Mixture Models for Incomplete Data

In this section, we briefly summarize how to implement Gaussian Mixture Models (GMM) when the data have missing features. This model will be exploited as the foundation for *PCKID* to learn a robust kernel function. For details, we address the interested reader to [10].

A GMM is used to model the probability density function (PDF) for given dataset. In a GMM, a data point $\mathbf{x}_i$ is assumed to be sampled from a multivariate Gaussian distribution $\mathcal{N}_k(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with probability π_k and $k \in [1, K]$, where K corresponds to the number of mixture components. Accordingly, the PDF of the data is modeled by a *mixture* of Gaussians, such that

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{1}$$

The maximum likelihood estimates for the parameters in this model can be approximated through the Expectation Maximization (EM) algorithm.

When the data have missing features, we assume that the elements in a data vector $\mathbf{x}_i$ can be partitioned into two components; one observed part $\mathbf{x}_i^{\mathrm{o}}$ and one missing part $\mathbf{x}_i^{\mathrm{m}}$ as explained in Sec. 2.1. Then, one can construct a binary matrix $\mathbf{O}_i$ by removing the rows from the identity matrix corresponding to the missing elements $\mathbf{x}_i^{\mathrm{m}}$, such that $\mathbf{x}_i^{\mathrm{o}} = \mathbf{O}_i\mathbf{x}_i$. Given the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$ for mixture component k, the mean and covariance matrix for the *observed* part of missingness pattern i is given by

$$\begin{aligned}\boldsymbol{\mu}_{k,i}^{\mathrm{o}} &= \mathbf{O}_i\boldsymbol{\mu}_k \\ \boldsymbol{\Sigma}_{k,i}^{\mathrm{o}} &= \mathbf{O}_i\boldsymbol{\Sigma}_k\mathbf{O}_i^T.\end{aligned}$$

Algorithm 1 EM algorithm for incomplete data GMM

1: Initialize $\hat{\boldsymbol{\mu}}_k^{(0)}$, $\hat{\boldsymbol{\Sigma}}_k^{(0)}$, $\hat{\pi}_k^{(0)}$ and $\hat{\gamma}_{i,k}^{(0)}$ for $k \in [1, K]$ and $i \in [1, N]$.
2: **while** not converged **do**
3: **E-Step:** Compute

$$\hat{\gamma}_{k,i}^{(\ell)} = \frac{\hat{\pi}_k^{(\ell)} \mathcal{N}\left(\mathbf{x}_i^{\mathrm{o}} | \hat{\boldsymbol{\mu}}_{k,i}^{\mathrm{o}(\ell)}, \hat{\boldsymbol{\Sigma}}_{k,i}^{\mathrm{o}(\ell)}\right)}{\sum_{j=1}^{K} \hat{\pi}_j^{(\ell)} \mathcal{N}\left(\mathbf{x}_i^{\mathrm{o}} | \hat{\boldsymbol{\mu}}_{j,i}^{\mathrm{o}(\ell)}, \hat{\boldsymbol{\Sigma}}_{j,i}^{\mathrm{o}(\ell)}\right)}$$

$$\hat{\mathbf{Y}}_{k,i}^{(\ell)} = \hat{\boldsymbol{\mu}}_k^{(\ell)} + \hat{\boldsymbol{\Sigma}}_k^{(\ell)} \hat{\mathbf{S}}_{k,i}^{\mathrm{o}(\ell)} \left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{(\ell)}\right)$$

4: **M-Step:** Compute the next model parameters, given by

$$\hat{\pi}_k^{(\ell+1)} = \frac{1}{N} \sum_{i=1}^{N} \hat{\gamma}_{k,i}^{(\ell)}$$

$$\hat{\boldsymbol{\mu}}_k^{(\ell+1)} = \frac{\sum_{i=1}^{N} \hat{\gamma}_{k,i}^{(\ell)} \hat{\mathbf{Y}}_{k,i}^{(\ell)}}{\sum_{i=1}^{N} \hat{\gamma}_{k,i}^{(\ell)}}$$

$$\hat{\boldsymbol{\Sigma}}_k^{(\ell+1)} = \frac{\sum_{i=1}^{N} \hat{\boldsymbol{\Omega}}_{k,i}^{(\ell)}}{\sum_{i=1}^{N} \hat{\gamma}_{k,i}^{(\ell)}},$$

where

$$\hat{\boldsymbol{\Omega}}_{k,i}^{(\ell)} = \hat{\gamma}_{k,i}^{(\ell)} \Bigg(\left(\hat{\mathbf{Y}}_{k,i}^{(\ell)} - \hat{\boldsymbol{\mu}}_k^{(\ell+1)}\right) \left(\hat{\mathbf{Y}}_{k,i}^{(\ell)} - \hat{\boldsymbol{\mu}}_k^{(\ell+1)}\right)^T + \left(\mathbf{I} - \hat{\boldsymbol{\Sigma}}_k^{(\ell)} \hat{\mathbf{S}}_{k,i}^{o(\ell)}\right) \hat{\boldsymbol{\Sigma}}_k^{(\ell)} \Bigg).$$

5: **end while**

By defining

$$\mathbf{S}_{k,i}^{\mathrm{o}} = \mathbf{O}_i^T \boldsymbol{\Sigma}_{k,i}^{\mathrm{o}\ -1} \mathbf{O}_i,$$

one can show that, under the MAR assumption, the EM procedure outlined in Alg. 1 will find the parameters that maximizes the likelihood function [10].

Note that, even though the notation in this paper allows for a unique missingness pattern for each data point $\mathbf{x}_i$, one missingness pattern is usually shared between several data points. Thus, to improve efficiency when implementing Alg. 1, one should sort the data points by missingness pattern such that parameters that are common across data points are calculated only once [10].

Diagonal covariance structure assumption. In some cases, when the dimensionality of the data is large compared to the number of data points, in combination with many missingness patterns, one could consider assuming a diagonal covariance structure for the GMM for computational efficiency and numerical stability when inverting covariance matrices. This will of course limit the models to not encode correlations between dimensions, but for some tasks it provides a good approximation that is a viable compromise when limited computational resources are available. In this case, covariance matrices are encoded in d-dimensional vectors, which simplifies the operations in Alg. 1.

Let $\widehat{\boldsymbol{\sigma}}_k$ be the vector of variances for mixture component k and let $\widehat{\mathbf{s}}_{k,i}$ be a vector with elements $\widehat{s}_{k,i}(\ell) = \frac{1}{\sigma_k(\ell)}$ if element ℓ of data point $\mathbf{x}_i$ is observed and

$\widehat{s}_{k,i}(\ell) = 0$ otherwise. Define

$$\widehat{\mathbf{y}}_{k,i} = \widehat{\boldsymbol{\mu}}_k + \widehat{\boldsymbol{\sigma}}_k \odot \widehat{\mathbf{s}}_{k,i} \odot (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k), \tag{2}$$

and

$$\boldsymbol{\omega}_{k,i} = \hat{\gamma}_{k,i} \left((\widehat{\mathbf{y}}_{k,i} - \widehat{\boldsymbol{\mu}}_k) \odot (\widehat{\mathbf{y}}_{k,i} - \widehat{\boldsymbol{\mu}}_k) + \widehat{\boldsymbol{\sigma}}_k - \widehat{\boldsymbol{\sigma}}_k \odot \widehat{\mathbf{s}}_{k,i} \odot \widehat{\boldsymbol{\sigma}}_k \right) \tag{3}$$

where $\odot$ denotes the Hadamard (element wise) product. Estimating the parameters with an assumption of diagonal covariance structure is then a matter of exchanging $\widehat{\mathbf{Y}}_{k,i}$ and $\boldsymbol{\Omega}_{k,i}$ with $\widehat{\mathbf{y}}_{k,i}$ and $\boldsymbol{\omega}_{k,i}$ respectively in Alg. 1.

2.3 Spectral clustering using Kernel PCA

Spectral clustering is a family of clustering algorithms, where the spectrum, i.e. the eigenvalues and eigenvectors, of some similarity matrix is exploited for clustering of data separated by non-linear structures [4, 14, 15, 22, 23]. Most spectral clustering algorithms employ a two-stage approach, with i) a non-linear feature generation step using the spectrum and ii) clustering by k-means on top of the generated features [14, 19]. Some have employed a strategy where the final clustering step is replaced by spectral rotations [15, 20] or by replacing both steps with kernel k-means [2], which is difficult to initialize. In this work, we employ the two stage approach where we use kernel PCA [18] to generate k-dimensional feature vectors, for then to cluster these using k-means.

Kernel PCA Kernel PCA implicitly performs PCA in some reproducing kernel Hilbert space $\mathcal{H}$ given a positive semidefinite kernel function $\kappa \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which computes inner products in $\mathcal{H}$. If we define a *kernel matrix*, $\mathbf{K}$, whose elements are the inner products $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$, this matrix is positive semidefinite, and may be decomposed as $\mathbf{K} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T$, where $\mathbf{E}$ is a matrix with the eigenvectors as columns and $\boldsymbol{\Lambda}$ is the diagonal eigenvalue matrix of $\mathbf{K}$. Then it can be shown that the k-dimensional projections onto the principal components in $\mathcal{H}$ is given by

$$\mathbf{Z} = \mathbf{E}_k \boldsymbol{\Lambda}_k^{\frac{1}{2}}, \tag{4}$$

where $\boldsymbol{\Lambda}_k$ consists of the k largest eigenvalues of $\mathbf{K}$ and $\mathbf{E}_k$ consists of the corresponding eigenvectors.

The traditional choice of kernel function is an RBF kernel, defined as

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \tag{5}$$

where the σ parameter defines the width of the kernel.

3 *PCKID* – A Probabilistic Cluster Kernel for Incomplete Data

In this paper, we propose a novel procedure to construct a kernel matrix based on models learned from data with missing features, which we refer to as *PCKID*. In particular, we propose to learn similarities between data points in an unsupervised fashion by fitting GMMs to the data with different initial conditions $q \in [1, Q]$ and a range of mixture components, $g \in [2, G]$ and combine the results using the posterior probabilities for the data points. That is, we define the kernel function as

$$\kappa_{PCKID}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{Z} \sum_{q=1}^{Q} \sum_{g=2}^{G} \boldsymbol{\gamma}_i^T(q, g) \boldsymbol{\gamma}_j(q, g), \tag{6}$$

where $\boldsymbol{\gamma}_i(q, g)$ is the posterior distribution for data point $\mathbf{x}_i$ under the model with initial condition q and g mixture components and Z is a normalizing constant. By using Alg. 1 to train the models, we are able to learn the kernel function from the inherent structures of the data, even when dealing with missing features. In this work, we use this kernel for spectral clustering.

The *PCKID* is able to capture similarities on both a local and a global scale. When a GMM is trained with many mixture components, each mixture component covers a small, *local* region in feature space. On the contrary, when the GMM is trained with a small number of mixture components, each mixture component covers a large, *global* region in feature space. Thus, if two data points are similar under models on all scales, they are likely to be similar, and will have a large value in the *PCKID*. This procedure of fitting models to the data on different scales, ensures robustness with respect to parameters, as long as Q and G are set sufficiently large. Thus, we are able to construct a kernel function that is robust with regards to parameter choice. This way of constructing a robust kernel is similar to the methodology used in ensemble clustering and recent work in spectral clustering [6]. However, such recent methods are not able to explicitly handle missing data.

According to the ensemble learning methodology [13,24], we build a powerful learner by combining multiple weak learners. Therefore, one does not need to run the EM algorithm until convergence, but instead perform just a few iterations[3]. This also has the positive side-effect of encouraging diversity, providing efficiency and preventing overfitting. To further enforce diversity, it is beneficial to use subsampling techniques to train different models on different subsets of the data and evaluate the complete kernel on the full dataset.

3.1 Initialization

For each mixture model that is trained, one needs to provide an initialization. Since we are fitting large models to data that in practice does not necessarily fit

[3] For instance, 10 iterations.

these models, the initialization needs to be reasonable in order to avoid computational issues when inverting covariance matrices. An initialization procedure that has been validated empirically for the *PCKID* is

1. Use mean imputation to impute missing values.
2. Draw K random data points from the input data and use them as initial cluster centers.
3. Run *one* k-means iteration to get initial cluster assignments and means.
4. Calculate the empirical covariance matrix from each cluster and calculate empirical prior probabilities for the mixture model based on the cluster assignments.

Data with imputed values is only used to be able to calculate initial means and covariances. *When training the model, data without imputed values is used.*

4 Experiments

4.1 Experiment setup

***PCKID* parameters** In order to illustrate that *PCKID* does not need any parameter tuning, the parameters are set to $Q = G = 30$ for all experiments. In order to increase diversity, each model in the ensemble is trained on a random subset of 50% of the whole dataset. The kernel is evaluated on the full dataset, once the models are trained. Each GMM is trained for 10 iterations with a diagonal covariance structure assumption.

Baseline methods For the baseline methods, missing data is handled with imputation techniques, in particular, i) zero imputation, ii) mean imputation iii) median imputation and (iv) most frequent value imputation. To produce a clustering result, each of these imputation techniques is coupled with i) k-means on the data and ii) spectral clustering using an RBF kernel, where the kernel function is calculated by (5).

Since no hyperparameters need to be tuned in in *PCKID*, the kernel width σ of the RBF is calculated with a rule of thumb. In particular, σ is set to 20% of the median pairwise distances in the dataset, as suggested in [8]. This is in agreement with unsupervised approaches, where labels are not known and cross validation on hyperparameters is not possible.

Performance metric In order to assess the performance of *PCKID*, its supervised clustering accuracy is compared with all baseline models. The supervised clustering accuracy is computed by

$$ACC = \max_{\mathcal{M}} \frac{\sum_{i=1}^{n} \delta\{y_i = \mathcal{M}(\widehat{y}_i)\}}{n}, \tag{7}$$

where y_i is the ground truth label, $\widehat{y}_i$ is the cluster label assigned to data point i and $\mathcal{M}(\cdot)$ is the label mapping function that maximizes the matching of the labels. This is computed using the Hungarian algorithm [9].

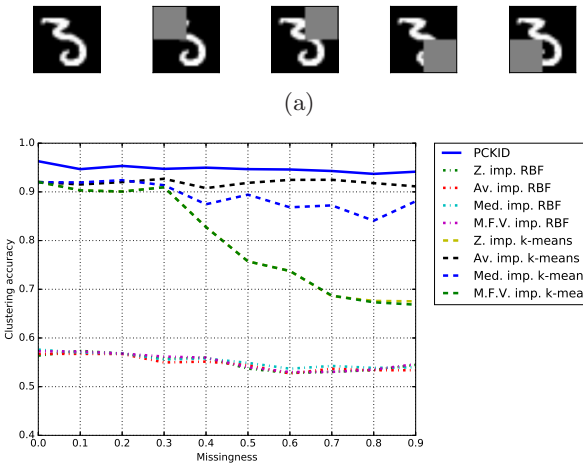
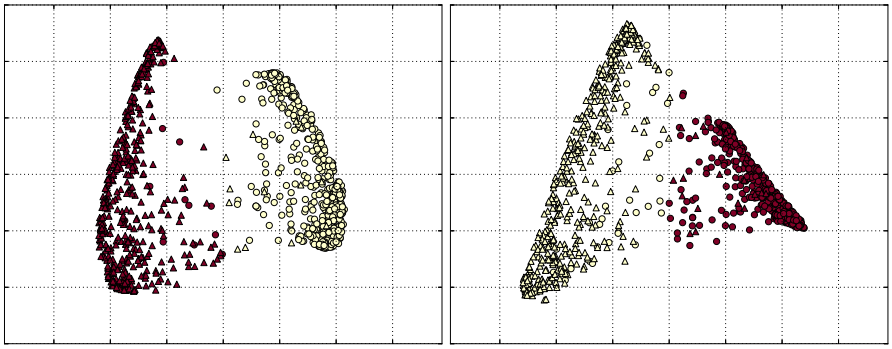


Fig. 1. (a): Example of missingness patterns. Gray pixels are considered missing. (b): Mean clustering accuracy as a function of the percentage of images with missing values.

Clustering setup Spectral clustering with k clusters is performed by mapping the data to a k dimensional empirical kernel space and clustering them with k-means as described in Sec. 2.3. For all methods, k-means is run 100 times. The final clustering is chosen by evaluating the k-means cost function and choosing the partitioning with the lowest cost. The number of clusters, k, is assumed known.

4.2 MNIST 5 vs. 6

In this experiment, subsets containing 1000 of the MNIST 5 and 6 images are clustered. The subsets consists of a balanced sample, i.e. there are approximately the same amount of images from each class. The images are unraveled to 784 dimensional vectors, which are used as the input to the algorithms. Missing data is generated by randomly choosing a proportion p_{m} of the images and removing one of the four quadrants in the image according to the MAR mechanism. These missingness patterns are illustrated in Fig. 1(a). In each test, we consider different probabilities of having missing quadrants, i.e. $p_{\text{m}} \in \{0.0, 0.1, 0.2, \ldots, 0.9\}$, Each method is run 30 times for each value of p_{m}, with a unique random subset of the data for each run. Since there are dimensions in the dataset where there is no variation between images, they are removed before training the GMMs. These are dimensions without information, and causes problems when inverting the covariance matrices. The number of dimensions with variance varies across the runs, since the subset from the dataset and the missingness is randomly sampled for each run. The number of dimensions with variance is approximately 500.



(a) (b)

Fig. 2. Example of embedding and clustering in kernel space with (a): No missingness, (b): 90% missingness. The marker indicates the true label, while the color indicates the clustering results.

Fig. 1(b) shows a plot of the mean clustering accuracy over the 30 runs versus the missingness proportion p_{m}. The proposed method outperforms the baseline methods for all p_{m}. Although the clustering accuracy declines slightly when the p_{m} increases, the results are quite stable.

Fig. 2(a)–Fig. 2(b) shows two dimensional representations using kernel PCA on *PCKID* with $p_{\mathrm{m}} = 0$ and $p_{\mathrm{m}} = 0.9$, respectively. The shape of the markers indicate ground truth class, while the color indicate the clustering result. It is interesting to see that although the plot with no missing data has a smoother structure, the overall topology seems to be very similar when $p_{\mathrm{m}} = 0.9$. The two classes seem to be less separable in the plot with more missing data, which is not surprising, given the numerical clustering results in Fig. 1(b).

When considering the approach of k-means directly on data with imputed values, we see that none of the imputation techniques perform as well as *PCKID*, although in this case mean imputation works reasonably well. To explain performance improvements as p_{m} increases, it is possible that the missingness patterns chosen for this experiment introduce some noise that provides a form of regularization that is beneficial to certain imputation techniques, or maybe the balance in the dataset is helping the mean of the observed values to not introduce bias towards one class. With median–, zero– and most frequent value imputation, the clustering accuracy starts to decline around $p_{\mathrm{m}} = 0.3$, with zero imputation and most frequent value imputation following almost exactly the same path. This is likely due to the nature of the data, where many of the dimensions actually contains zeros in most of the images. The most frequent value in most dimensions will then be zero.

Spectral clustering using an RBF kernel completely fails in this experiment, which is probably due to a sub-optimal kernel width. However, this illustrates the difficulty with an unsupervised problem, where no prior information is given,

Table 1. Average clustering accuracy over 30 runs for different combinations of classes in the Hardangervidda dataset. The best results are marked in bold. The baseline methods are: ZI (zero imputation), AI (average imputation), MI (median imputation) and MFVI (most frequent value imputation), combined with either k-means or spectral clustering using an RBF kernel.

Classes	*PCKID*	Spectral clustering, RBF				k-means			
		ZI	AI	MI	MFVI	ZI	AI	MI	MFVI
2-3	0.580	0.610	0.610	0.624	**0.627**	0.601	0.601	0.601	0.605
2-4	0.536	0.663	0.663	0.663	**0.674**	0.591	0.591	0.590	0.597
2-5	0.661	0.589	0.589	0.598	0.605	**0.671**	**0.671**	0.663	0.652
2-6	**0.712**	0.578	0.578	0.571	0.594	0.672	0.672	0.664	0.639
2-7	**0.868**	0.519	0.519	0.516	0.501	0.854	0.854	0.858	0.862
3-4	0.698	0.505	0.505	0.505	0.511	0.697	0.697	0.711	**0.722**
3-5	**0.563**	0.521	0.521	0.511	0.516	0.534	0.534	0.540	0.540
3-6	**0.620**	0.565	0.565	0.562	0.564	0.521	0.521	0.519	0.523
3-7	**0.933**	0.501	0.501	0.726	0.522	0.577	0.577	0.599	0.603
4-5	0.764	0.517	0.517	0.512	0.510	0.839	0.839	0.847	**0.848**
4-6	**0.897**	0.517	0.517	0.547	0.547	**0.897**	**0.897**	0.894	0.880
4-7	**0.931**	0.550	0.550	0.547	0.534	0.687	0.687	0.687	0.718
5-6	**0.740**	0.623	0.623	0.644	0.672	0.554	0.554	0.602	0.606
5-7	**0.956**	0.687	0.687	0.667	0.698	0.706	0.706	0.706	0.706
6-7	**0.970**	0.767	0.767	0.752	0.696	0.759	0.759	0.759	0.670

making cross-validation virtually impossible without expertise knowledge on the data.

4.3 Land cover clustering

In this experiment, we cluster pixels in high resolution land cover images contaminated with clouds, also used for classification in [16, 17]. The data consists of three Landsat ETM+ images covering Hardangervidda in southern Norway, in addition to elevation and slope information. With 6 bands in each image, the total dimensionality of the data is 20. In this dataset, a value is considered missing if a pixel in an image is contaminated by either clouds or snow/ice. For details on how the dataset is constructed, see [16].

The pixels in the image are labeled as one of 7 classes: 1) *water*, 2) *ridge*, 3) *leeside*, 4) *snowbed*, 5) *mire*, 6) *forest* and 7) *rock*. In this experiment, we exclude the water class, since it is easy to separate from the other classes in the Norwegian mountain vegetation. To investigate how the *PCKID* handle the different combination of classes, we restrict the analysis to pairwise classes. Each dimension is standardized on the observed data.

The average clustering accuracy for each combination of the chosen classes is reported in Tab. 1. The average is computed over 30 runs of each algorithm. We see that *PCKID* seems to perform better for most class pairs. Although it might struggle with some classes, most notably class 2. For the class pair 3-5, *PCKID* wins with a clustering accuracy of 0.563, which is not much better than random chance in a two class problem. It is however worth to note that the classes labels are set according the vegetation at the actual location, which is not necessarily

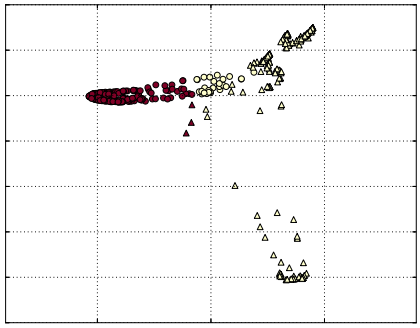


Fig. 3. Example of mapping for the *forest–rock* class pair. Colors indicate clustering, while the shape of the marker indicate the ground truth label.

the group structure reflected in the data. The class combinations where *PCKID* really outperforms the other methods seems to be when class 7 (rocks) is present in the data, where we improve performance by up to 25 percentage points with regards to the baseline methods.

Fig. 3 shows an example of a mapping for the *forest–rock* class pair, where it seems like the *rock* class, as defined by the ground truth, actually consists of two separate structures in the KPCA embedding using *PCKID*. This demonstrates the power of *PCKID*s ability to adapt to the inherent structures in the data.

5 Conclusion

In this paper, we have proposed *PCKID*, a novel kernel function for spectral clustering, designed to i) explicitly handle incomplete data and ii) be robust with regards to parameter choice. By combining posterior distributions of Gaussian Mixture Models for incomplete data on different scales, *PCKID* is able to learn similarities on the data manifold, yielding a kernel function without any *critical* hyperparameters to tune. Experiments have demonstrated the strength of our method, by improved clustering accuracy compared to baseline methods, while keeping parameters fixed for all experiments.

Acknowledgments This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines*.

References

1. Chechik, G., Heitz, G., Elidan, G., Abbeel, P., Koller, D.: Max-margin classification of data with absent features. JMLR 9(Jan), 1–21 (2008)
2. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means: spectral clustering and normalized cuts. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 551–556. ACM (2004)
3. Dixon, J.K.: Pattern recognition with partly missing data. IEEE Transactions on Systems, Man, and Cybernetics 9(10), 617–621 (1979)

4. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern recognition 41(1), 176–190 (2008)
5. Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an em approach. Advances in neural information processing systems pp. 120–120 (1994)
6. Izquierdo-Verdiguier, E., Jenssen, R., Gómez-Chova, L., Camps-Valls, G.: Spectral clustering with the probabilistic cluster kernel. Neurocomputing 149, 1299–1304 (2015)
7. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters 31(8), 651–666 (2010)
8. Jenssen, R.: Kernel entropy component analysis. IEEE transactions on pattern analysis and machine intelligence 32(5), 847–860 (2010)
9. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly 2(1-2), 83–97 (1955)
10. Lin, T.I., Lee, J.C., Ho, H.J.: On fast supervised learning for normal mixture models with missing information. Pattern Recognition 39(6), 1177–1187 (2006)
11. Marlin, B.M.: Missing data problems in machine learning. Ph.D. thesis, University of Toronto (2008)
12. Mojirsheibani, M., Montazeri, Z.: Statistical classification with missing covariates. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69(5), 839–857 (2007)
13. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine learning 52(1-2), 91–118 (2003)
14. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems. pp. 849–856 (2001)
15. Nie, F., Zeng, Z., Tsang, I.W., Xu, D., Zhang, C.: Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. IEEE Transactions on Neural Networks 22(11), 1796–1808 (2011)
16. Salberg, A.B.: Land cover classification of cloud-contaminated multitemporal high-resolution images. IEEE Transactions on Geoscience and Remote Sensing 49(1), 377–387 (2011)
17. Salberg, A.B., Jenssen, R.: Land-cover classification of partly missing data using support vector machines. International journal of remote sensing 33(14), 4471–4481 (2012)
18. Schölkopf, B., Smola, A., Müller, K.R.: Kernel principal component analysis. In: International Conference on Artificial Neural Networks. pp. 583–588. Springer (1997)
19. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence 22(8), 888–905 (2000)
20. Stella, X.Y., Shi, J.: Multiclass spectral clustering. In: ICCV. pp. 313–319 (2003)
21. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Fourth Edition. Academic Press, 4th edn. (2008)
22. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing 17(4), 395–416 (2007)
23. Yang, Y., Xu, D., Nie, F., Yan, S., Zhuang, Y.: Image clustering using local discriminant models and global integration. IEEE Transactions on Image Processing 19(10), 2761–2773 (2010)
24. Zimek, A., Gaudet, M., Campello, R.J., Sander, J.: Subsampling for efficient and effective unsupervised outlier detection ensembles. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 428–436. ACM (2013)