2011 10th International Conference on Machine Learning and Applications

Predictive Subspace Clustering

Brian McWilliams and Giovanni Montana*
Statistics Section, Department of Mathematics
Imperial College
London, SW7 2AZ
Email: brian.mcwilliams07@imperial.ac.uk, g.montana@imperial.ac.uk*
*Corresponding author.

***Abstract*—The problem of detecting clusters in high-dimensional data is increasingly common in machine learning applications, for instance in computer vision and bioinformatics. Recently, a number of approaches in the field of subspace clustering have been proposed which search for clusters in subspaces of unknown dimensions. Learning the number of clusters, the dimension of each subspace, and the correct assignments is a challenging task, and many existing algorithms often perform poorly in the presence of subspaces that have different dimensions and possibly overlap, or are otherwise computationally expensive. In this work we present a novel approach to subspace clustering that learns the numbers of clusters and the dimensionality of each subspace in an efficient way. We assume that the data points in each cluster are well represented in low-dimensions by a PCA model. We propose a measure of predictive influence of data points modelled by PCA which we minimise to drive the clustering process. The proposed predictive subspace clustering algorithm is assessed on both simulated data and on the popular Yale faces database where state-of-the-art performance and speed are obtained.**

I. Introduction

A growing number of modern machine learning applications require algorithms for the automatic discovery of naturally occurring clusters of very high-dimensional observations, such as digital images or gene expressions. When dealing with such data, often a plausible assumption is that within each cluster, the true dimensionality of the data is much smaller and the clusters exist and can be found only in low-dimensional subspaces. In the most general case, the subspaces that identify these data partitions will not have the same dimensions and may even overlap [**?**].

A typical application is encountered in the domain of facial recognition using digital images. A set of images representing a single individual and taken under some particular lighting conditions generally lie in a low-dimensional linear subspace that captures the distinguishing features unique to that person [**?**]. When a collection of images representing multiple individuals is available, the clusters of identical faces can be recovered by inferring each individual subspace. Analogous examples can be found in the area of computer vision, where there is the need to cluster motion trajectories from video sequences [**?**], and in genomics, where biological samples are partitioned based on their gene expression signatures [**?**].

In Section II we introduce more formally the problem of detecting subspaces under which the data cluster, and briefly review state-of-the-art algorithms for subspace clustering and their limitations which relate to the difficult problem of identifying the number and dimensionality of the subspaces.

In Section III we present our approach based on optimally fitting multiple PCA models to the data. We define a notion of *predictive influence* of an observation under a PCA model, and use this to design an objective function for data partitioning so that each recovered cluster contains observations which are similar to each other in a predictive sense. In this respect, our algorithm performs *predictive* subspace clustering and overcomes the limitations of current approaches. Extensive Monte Carlo simulation results and an application to real data, discussed in Section IV, demonstrate the state-of-the-art performance of our algorithm. We conclude in Section V.

II. Subspace clustering

We assume to have observed N points, $\{\boldsymbol{x}_i\}_1^N$, where each $\boldsymbol{x}_i \in \mathbb{R}^{1\times P}$ and the dimension P is usually very large. Each point is assumed to belong to one of K non-overlapping clusters, $\{\mathcal{C}_k\}_1^K$. We further assume that the points in the k^{th} cluster lie in a R_k-dimensional subspace, $\mathcal{S}_k$ where $R_k << P$. Each subspace $\mathcal{S}_k$ is defined in the following way

$$\mathcal{S}_k = \{\boldsymbol{x}_i : \boldsymbol{x}_i = \boldsymbol{\mu}^{(k)} + \boldsymbol{u}_i^{(k)}\boldsymbol{V}^{(k)\top}\} \tag{1}$$

with $i \in \mathcal{C}_k$ and $k = 1,\dots,K$, where $\boldsymbol{V}^{(k)} \in \mathbb{R}^{P\times R_k}$ is a basis for $\mathcal{S}_k$ whose columns are restricted to be mutually orthonormal. The point $\boldsymbol{u}_i^{(k)} \in \mathbb{R}^{R_k}$ is the low dimensional representation of $\boldsymbol{x}_i$ and $\boldsymbol{\mu}^{(k)} \in \mathbb{R}^P$ is an arbitrary point in $\mathcal{S}_k$, typically chosen to be $\mathbf{0}$.

When only one cluster exists, i.e. $K = 1$, a subspace of this form can be estimated by fitting a PCA model, which provides the best low-rank linear approximation of the original data. One way of achieving this is by estimating a set of R mutually orthonormal vectors $[\boldsymbol{v}^{(1)},\dots,\boldsymbol{v}^{(R)}]$ which minimize the L_2 reconstruction error, defined as:

$$\sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}_i \sum_{r=1}^{R} \boldsymbol{v}^{(r)}\boldsymbol{v}^{(r)\top}\|^2. \tag{2}$$

Defining a matrix $\boldsymbol{X} \in \mathbb{R}^{N\times P}$ with rows $\boldsymbol{x}_i$, which we assume to be mean-centred, the vectors which minimize Eq. (2) are obtained by computing the singular value decomposition (SVD) of $\boldsymbol{X}$, given by $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$. Here, $\boldsymbol{U} = [\boldsymbol{u}^{(1)},...,\boldsymbol{u}^{(N)}] \in \mathbb{R}^{N\times N}$ and $\boldsymbol{V} = [\boldsymbol{v}^{(1)},...,\boldsymbol{v}^{(P)}] \in \mathbb{R}^{P\times P}$ are orthonormal matrices whose columns are the

978-0-7695-4607-0/11 $26.00 © 2011 IEEE
DOI 10.1109/ICMLA.2011.117

left and right singular vectors of $\boldsymbol{X}$, respectively. $\boldsymbol{\Lambda} = \text{diag}(\lambda^{(1)}, \ldots, \lambda^{(N)}) \in \mathbb{R}^{N \times P}$ is a diagonal matrix whose entries are the singular values of $\boldsymbol{X}$ in descending order.

When the data partition into more than one cluster, i.e. $K > 1$, but the cluster assignments are known, a subspace of form (1) can be estimated by fitting a PCA model independently in each cluster. However, since the cluster assignments are generally unknown, the problem of subspace clustering consists in the simultaneous identification of the number of clusters, K, the subspaces $\{\mathcal{S}_i\}_1^K$ and the cluster assignments $\{\mathcal{C}_i\}_1^K$.

As noted earlier, there are several fundamental difficulties associated with this problem: (a) identifying the true subspaces is dependent on recovering the true clusters and vice-versa; (b) subspaces can intersect at several locations which causes difficulties when attempting to assign points to subspaces at these intersections, and standard clustering techniques such as K-means may not be suitable; (c) the subspace parameters and the cluster assignments are dependent on both number of clusters, K and the dimensionality of their respective subspaces, which pose difficult estimation challenges.

Recently, a variety of approaches have been proposed to solve the subspace clustering problem, although the problem of inferring the number of clusters K and the varying dimensionality of each subspace has remained partially unsolved. Among these methods, several are based on generalising the K-means algorithm to *K-subspaces* [**?**], [**?**]. These methods iteratively fit PCA models to the data and assign points to clusters until the PCA reconstruction error (Eq. 2) in each cluster is minimised.

Although the approach based on minimising the within-cluster PCA reconstruction error is simple and has shown promising results, it is also prone to over-fitting. For instance, the data may be corrupted by noise or lie on the intersection between subspaces and so points within clusters may be geometrically far apart. Since the PCA reconstruction error is not robust to outliers, such points may bias the estimated subspace which is a fundamental limitation for assigning points to clusters. Furthermore, each subspace may have a different intrinsic dimensionality. The PCA reconstruction error decreases monotonically as the dimensionality increases, so points may be wrongly assigned to the cluster with the largest dimensionality. Such an approach therefore limits the number of dimensions to be the same in each cluster.

Another class of subspace clustering algorithms are based on computing a measure of distance between each pair of points which in some way captures the notion that points may lie on different subspaces. The distances are then used to construct an affinity matrix which is partitioned using standard spectral clustering techniques [**?**].

There have been several successful approaches to defining such a distance measure. The method of Generalized PCA (GPCA) [**?**] fits K, order-R_k polynomials to the data and measures the distances between the gradient of the polynomials computed at each point. Sparse subspace clustering (SSC) [**?**] obtains a local representation of the subspace at each point as a sparse weighted sum of all other points; this is obtained by minimising the reconstruction error subject to a constraint on the L_1 norm of the weights so that the few non-zero weights correspond to points lying on the same subspace. Spectral curvature clustering (SCC) [**?**] constructs a multiway distance between randomly sampled groups of points by measuring the volume of the of the simplex formed by each group. Spectral local best flats (SLBF) [**?**] estimates a local subspace for each point by fitting a PCA model to its nearest neighbours. SLBF then computes pairwise distances between the locally estimated subspaces corresponding to each point.

Although these spectral methods typically far outperform K-subspaces and achieve state-of-the art results in the domains of motion segmentation and image clustering, they introduce their own set of limitations. Computing local subspaces for each point can be computationally intensive and requires additional tunable parameters. Furthermore, the model parameters K and all R_k must typically still be fixed in advance although some methods exist to reliably estimate the number of clusters [**?**].

III. A PREDICTIVE PCA APPROACH TO SUBSPACE CLUSTERING

A. Overview

In this work we propose a novel approach to solving the subspace clustering problem which maintains some similarity to the K-subspaces algorithm. As with K-subspaces, we iteratively fit cluster-wise PCA models and reassign points to clusters until a certain optimality condition is met. However, rather than trying to minimise the residuals under the individual PCA models, we introduce an objective function that exploits the predictive nature of the PCA problem in Eq. 2 in a way that makes it particularly robust to noise and outliers. This also enables the resulting algorithm to learn both the number of clusters and the dimensionality of each subspace. An outline of the main contributions is in order.

First, we propose an efficient solution for detecting influential observations under a PCA model. An influential observation is generally defined as a point which exerts a larger effect on the estimated parameters compared to other points in the model. A common and robust method to detect influence is by examining the effect of the parameters when the model is estimated by leaving out each observation in turn. In the context of ordinary least squares (OLS) this is equivalent to computing the leave-one-out (LOO) estimate of the regression coefficients with respect to each observation. The LOO prediction error is then evaluated using the remaining observation. Influential observations can be identified as those with a large LOO prediction error relative to other observations.

In the context of OLS, the LOO prediction error has an analytic form which can be efficiently evaluated for all N observations at the expense of a single OLS model fit, known as the Predicted REsidual Sum of Squares (PRESS) [**?**]. In the context of PCA, the PRESS has also been used for both model selection and detecting influential observations [**?**]. However, evaluating the exact LOO error requires the SVD

to be re-computed N times, and would be impractical for use in any iterative algorithm. Here we propose a closed-form solution for computing an approximated PCA PRESS, where the approximation error is negligible for practical purposes.

Second, armed with this analytical expression for the PCA PRESS, we propose a notion of *predictive influence* that an observation exerts on the PCA model, which can be used to detect influential observations in PCA. Compared to the standard residual error in Eq. (2), this quantity provides a robust measure that is less prone to noise and over-fitting.

Finally, building on this notion of predictive influence, we develop an algorithm for subspace clustering that discovers clusters characterised by their own PCA model. The optimality criterion we embrace is such that total within-cluster predictive influence is minimised. As we will see, this provides a more robust alternative to minimising the reconstruction error within each cluster and has several other advantages. Since this approach makes direct use of out-of-sample prediction errors, model selection can naturally be incorporated into the subspace clustering framework.

B. A closed-form PCA PRESS measure

In order to identify which observations are influential under a given PCA model, we first consider the exact LOO reconstruction error using the first principal component. This quantity has the following form

$$J = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}_i \boldsymbol{v}_{-i}\boldsymbol{v}_{-i}^\top\|^2,$$

where each $\boldsymbol{v}_{-i}$ is the first right singular vector of the SVD estimated using all but the i^{th} observation of $\boldsymbol{X}$. For R principal components, J can be estimated by computing the LOO estimates of the first R singular vectors. Therefore, computing J requires N SVD computations which is expensive when either N or P is large.

We instead assume that when the number of samples, N is large the estimate of the principal subspace, $\boldsymbol{v}$ does not change much if we estimate the SVD using N or $N-1$ observations. In other words we assume that $\boldsymbol{v}_{-i} \approx \boldsymbol{v}$ and therefore $\boldsymbol{x}_i\boldsymbol{v}_{-i} \approx \boldsymbol{x}_i\boldsymbol{v}$ for $i = 1, \ldots, N$. This is known as the Projection Approximation Subspace Tracking (PAST) approximation [**?**]. Using this approximation and defining $d_i = \boldsymbol{x}_i\boldsymbol{v}$, we can express the PRESS in terms of the i^{th} LOO errors, $\boldsymbol{e}_{-i}$, as a quadratic function of $\boldsymbol{v}_{-i}$ in the following way

$$J = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{e}_{-i}\|^2 = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - d_i\boldsymbol{v}_{-i}^\top\|^2. \tag{3}$$

Now $\boldsymbol{v}_{-i}$ can be obtained through least squares estimation as $\boldsymbol{v}_{-i} = \left(\boldsymbol{d}_{-i}^\top\boldsymbol{d}_{-i}\right)^{-1}\left(\boldsymbol{X}_{-i}^\top\boldsymbol{d}_{-i}\right)$.

Since computing the LOO solution has been reduced to an OLS problem, we can derive a recursive expression for $\boldsymbol{v}_{-i}$ in terms of the original eigenvector $\boldsymbol{v}$ using the Sherman-Morrison theorem

$$\boldsymbol{v}_{-i} = \boldsymbol{v} - \frac{\left(\boldsymbol{x}_i^\top - d_i\boldsymbol{v}\right)Dd_i}{1-h_i},$$

where $h_i = d_iDd_i$ and $D = \left(\boldsymbol{d}^\top\boldsymbol{d}\right)^{-1}$. Now, using this expression for $\boldsymbol{v}_{-i}$ in Eq (3) we obtain the i^{th} PRESS error for PCA as

$$\boldsymbol{e}_{-i} = \frac{\boldsymbol{e}_i}{1-h_i},$$

where $\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{x}_i\boldsymbol{v}\boldsymbol{v}^\top$ is the i^{th} reconstruction error. From this expression, it can be seen that the i^{th} leave one out error can be written in terms of the i^{th} reconstruction error and quantities estimated by performing PCA without any explicit LOO steps.

Since the contribution of subsequent latent factors to the reconstruction error is additive, we can easily obtain the LOO error for $R > 1$ PCA components. This is achieved by simply computing the PRESS errors obtained using $r = 2, \ldots, R$ separately in the same way as for $r = 1$ and adding their contributions in the following way

$$\boldsymbol{e}_{-i}^{(R)} = \sum_{r=1}^{R} \frac{\boldsymbol{e}_i^{(r)}}{1-h_i^{(r)}} - (R-1)\boldsymbol{x}_i,$$

where $\boldsymbol{e}_i^{(r)} = \boldsymbol{x}_i - \boldsymbol{x}_i\boldsymbol{v}^{(r)}{\boldsymbol{v}^{(r)}}^\top$.

Finally, the full PRESS for the PCA reconstruction error for a PCA model with R components is given by

$$J^{(R)} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{e}_{-i}^{(R)}\|^2. \tag{4}$$

which depends only on quantities estimated using a single PCA model fit. Since $\boldsymbol{v}_{-i}$ is an eigenvector of $\sum_{j\neq i}^{N}\boldsymbol{x}_j$, it is constrained to be mutually orthonormal to any subsequently estimated eigenvectors. However, enforcing such a constraint on $\boldsymbol{v}_{-i}$ involves a re-normalization operation which breaks the linear recursive relationship between $\boldsymbol{v}$ and $\boldsymbol{v}_{-i}$. Therefore, approximating each LOO estimate of $\boldsymbol{v}_{-i}$ using least squares relies on relaxing this constraint which induces a small deviation from orthonormality.

Since computing the LOO errors for all $i = 1, \ldots, N$ relies on the eigenvector $\boldsymbol{v}$, estimated using all of the data, the computation of $\boldsymbol{v}_{-i}$ is effectively applying the PAST algorithm in reverse for a single iteration between N and $N-1$. It has been shown that the deviation from orthonormality in the PAST solution when initialised with a random unit vector, depends on the number of observations, N as $O(\frac{1}{N^2})$ [**?**]. As the number of samples (and thus iterations) grows, the estimates converge to the true eigenvectors.

C. A measure of predictive influence for PCA

We now define a measure of influence for an observation, $\boldsymbol{x}_i$ on the LOO prediction error of the PCA model. We define this as the gradient of the PCA PRESS in Eq. (4) with respect to $\boldsymbol{x}_i$ which we evaluate using the chain rule,

$$\frac{\partial J}{\partial \boldsymbol{x}_i} = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{e}_{-i}\|^2 = \frac{1}{2}\boldsymbol{e}_{-i}\frac{\partial}{\partial \boldsymbol{x}_i}\boldsymbol{e}_{-i}.$$

The *predictive influence* $\boldsymbol{\pi}(\boldsymbol{x}_i) \in \mathbb{R}^{P\times 1}$ of a point, $\boldsymbol{x}_i$ under a PCA model then has the following form:

$$\boldsymbol{\pi}(\boldsymbol{x}_i) = \boldsymbol{e}_{-i}^{(R)} \left(\sum_{r=1}^{R} \frac{\left(\boldsymbol{I}_p - \boldsymbol{v}^{(r)} \boldsymbol{v}^{(r)\top} \right)}{\left(1 - h_i^{(r)} \right)} - (R-1) \right).$$

The predictive influence measures the sensitivity of the prediction error in response to an incremental change in in the observation $\boldsymbol{x}_i$. The rate of change of the PRESS at this point is given by the magnitude of the predictive influence vector, $\|\boldsymbol{\pi}(\boldsymbol{x}_i)\|^2$. If the magnitude of the predictive influence is large, this implies a small change in the observation will result in a large change in the prediction error relative to other points. In this case, removing such a point from the model would cause a large improvement in the prediction error. We can then identify the most influential observations as those for which the increase in the PRESS is larger relative to other observations.

D. The Predictive Subspace Clustering (PSC) algorithm

The proposed algorithm relies on the following observation. If the cluster assignments $\{\mathcal{C}_k\}_1^K$ were known and a PCA model was fit to the data in each cluster, then the predictive influence of a point $\boldsymbol{x}_i$ belonging to cluster $\mathcal{C}_k$ would be small when evaluated using the correct PCA model for that cluster, and would be larger when using any of the remaining $K-1$ PCA models. In this respect, the predictive influence provides a robust and easy to compute goodness of fit measure that can be used to drive the clustering process.

The objective of the clustering algorithm is to partition the N observations $\boldsymbol{x}_i$ into one of K non-overlapping clusters such that each cluster contains exactly N_k observations and $\sum_{k=1}^{K} N_k = N$. Assuming that K is known, we recover the partitioning by ensuring each point is assigned to the cluster for which it exerts the smallest predictive influence relative to all other PCA models. The objective function to be minimized is defined as the sum of within-cluster predictive influences,

$$C = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \|\boldsymbol{\pi}_k(\boldsymbol{x}_i)\|^2, \tag{5}$$

where $\boldsymbol{\pi}_k(\boldsymbol{x}_i)$ is the predictive influence of a point $\boldsymbol{x}_i$ under the k^{th} PCA model. It is clear that if the expression in Eq. (5) is minimised the prediction error for each PCA model will be minimised since all points in each cluster will exert minimum predictive influence.

Minimising Eq. (5) involves simultaneously determining the true partitioning of the observations and estimating PCA model parameters for those K partitions. Since the true partitioning is unknown, there is no analytic solution to this problem and we must resort to an iterative procedure. The problem of estimating both the subspaces and the optimal cluster assignments can be attacked by considering the two related optimisation problems:

1) Given K subspaces with parameters, $\boldsymbol{V}^{(1)}, \ldots, \boldsymbol{V}^{(K)}$ and keeping these fixed, recover the cluster assignments which solve
$$\min_{\{\mathcal{C}_1,\ldots,\mathcal{C}_K\}} C. \tag{6}$$
2) Given a set of cluster assignments, $\{\mathcal{C}_1, ..., \mathcal{C}_K\}$ and keeping these fixed, estimate the parameters of the K subspaces which solve
$$\min_{\boldsymbol{V}^{(1)},\ldots,\boldsymbol{V}^{(K)}} C. \tag{7}$$

Obtaining cluster assignments by solving (6) changes the PCA model parameters obtained as a result of solving (7) and *vice versa*, therefore these objective functions must be solved iteratively. We propose an algorithm which minimizes Eq. (5) by alternately solving (6) and (7). At each iteration, the PSC algorithm follows two main steps which are outlined below.

Given an initial partitioning, $\{\mathcal{C}_k^{(0)}\}_1^K$ and the associated PCA models estimated using each cluster, which provides the parameters $\{\boldsymbol{V}_k^{(0)}\}_1^K$, the PSC algorithm is initialised by computing the predictive influences, $\boldsymbol{\pi}_k^{(0)}(\boldsymbol{x}_i)$ for $k = 1, \ldots, K$ and $i = 1, \ldots, N$. At each iteration $\tau = 1, 2, \ldots$, the following two steps are repeated until convergence: (a) assign points to clusters such that

$$\mathcal{C}_k^{(\tau)} \leftarrow \left\{ i : \min_k \|\boldsymbol{\pi}_k^{(\tau-1)}(\boldsymbol{x}_i)\|^2 \right\}.$$

and (b) re-estimate all PCA models, obtain the parameter set $\{\boldsymbol{V}_k^{(\tau)}\}_1^K$, and update each predictive influence $\boldsymbol{\pi}_k^{(\tau)}(\boldsymbol{x}_i)$ for all $k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, N$.

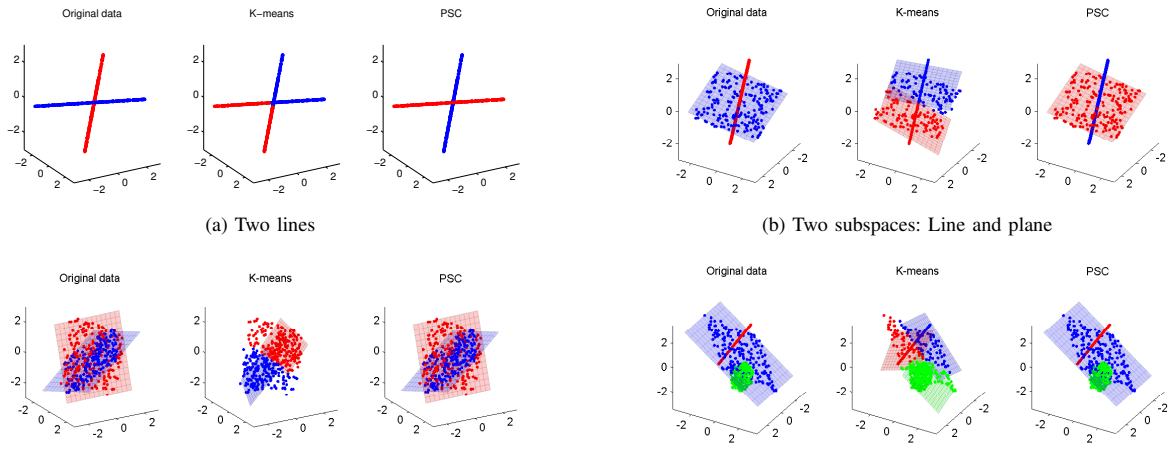
E. Model selection

The PRESS statistic provides a robust method for efficiently evaluating the fit of the PCA models within our framework. Straightforward extensions of the basic algorithm allow us to identify the optimal number of clusters, K, and the dimensionality of each subspace, $\{R_k\}_1^K$.

Assuming K is known at each iteration τ, using all data points in each cluster k, we evaluate all PCA PRESS statistics as in Eq. (4) using all values of each $R_k^{(\tau)}$ in a set $\mathcal{R} \equiv \{1, \ldots, R_{max}\}$. We select each $R_k^{(\tau)}$, that is the optimal subspace dimension in cluster k at the current iteration, to be the one that minimises the PRESS at that iteration.

The number of subspaces, K is estimated using a scheme in which we dynamically add and remove clusters from the model. If a cluster is not supported by the data, it is allowed to drop out of the model naturally. We add a cluster by identifying the cluster which exhibits the largest PRESS after convergence and dividing its observations between two new clusters, thereby increasing the number of clusters to $K+1$. This process continues until the overall PRESS is no longer decreased by adding a new cluster. Furthermore, the splitting operation makes the PSC algorithm less susceptible to local optimal solutions as it performs a more thorough search of the possible cluster configurations.

Our PRESS statistic for PCA allows both K and R_k to be efficiently learned from the data. Given that the SVD at each iteration and for each cluster has been computed up to



(a) Two lines

(b) Two subspaces: Line and plane

(c) Two planes

(d) Three subspaces: Line, plane and sphere

Fig. 1. Example results of clustering data belonging to several different subspaces using K-means and PSC. In these examples PSC consistently recovers the true cluster assignments and estimates the subspaces correctly.

dimension R_{max}, the additional computational effort required to evaluate all R_{max} values of the PRESS is negligible since all the quantities required to compute the PRESS are obtained via the SVD. Furthermore, PSC only requires the user to specify the single parameter, R_{max}. Since each one of the R_k values is typically much smaller than P, we can set R_{max} to be small relative to P so the computation of the full, P-dimensional SVD is not necessary.

IV. Experimental results

A. Simulated data

We first present simple simulated examples to illustrate the type of clusters that can be detected by the proposed PSC algorithm. We generate clusters of 100 data points which are distributed uniformly on a one, two or three-dimensional linear subspace embedded in three-dimensional space. To define each subspace, we generate a set of R_k orthonormal basis vectors each of dimension $P = 3$, where each element is sampled from a standard normal distribution. For each cluster we then sample 100 R_k-dimensional points from a uniform distribution which are then projected onto its corresponding subspace.

In Figure 1 we consider four simulation scenarios which consist of points which lie on: (a) two straight lines, (b) a straight line and a plane, (c) two planes and, finally, (d) a straight line, a plane and a sphere of unit radius. In each of these cases, we show the original data points in P dimensions, the clustering assignments using K-means clustering in the original dimensions, and the clustering assignment using PSC. It can be noted that the subspaces intersect so points belonging to different clusters may lie close to each other. We apply the K-means algorithm, which uses the Euclidean distance between points, directly to the 3-D data and as expected it consistently fails to recover the true clusters. On the other hand, PSC correctly recovers both the true clusters and the intrinsic dimensionality of the subspaces.

We also consider an additional scenario, (e), where $P = 200$ and $K = 4$. Here, the clusters consist of points which lie uniformly on a 5-D hyperplane, 4-D hypersphere and two lines generated as before. Using all five scenarios, we carry out a study whereby, in each case, we simulate 200 random data sets and compare the performance of five competing subspace clustering algorithms: PSC, GPCA, SCC, SSC and SLBF.

Table I reports on both the mean percentage of incorrectly clustered points using the Rand index and computation time (in seconds). In scenarios (a)-(c) all competing methods achieve close to perfect clustering accuracy except GPCA which performs poorly under the scenario (b) where the intrinsic dimensionality of the clusters is different. All the competing methods perform poorly under scenario (d) where the clusters exist on three subspaces of different dimensions. Even in low-dimensions, GPCA, SSC and SLBF require at least an order of magnitude more computation time compared to PSC. In the high-dimensional scenario (e), SLBF and SSC incur further computational cost as P, K and N increases. GPCA cannot be applied in such high-dimensional settings. Our PSC algorithm accurately recovers the clusters in all settings with little computational cost. This is due to the ability to automatically learn the dimensionality of each subspace.

B. An application to clustering face images

We apply the PSC algorithm to a common benchmark dataset, the Yale faces B database [**?**]. The data consists of frontal images of 10 individual faces taken under 64 different illumination conditions, so $N = 640$. Each image has dimensions $P = 120 \times 160 = 19200$ pixels. We represent each image by a vector and concatenate all the images so that each individual is represented by a matrix $\boldsymbol{X} \in \mathbb{R}^{640 \times 19200}$. Figure 2 shows the image of all 10 subjects under ambient lighting conditions. It is known that a set of images of objects in a fixed pose under varying lighting conditions can be well approximated by a low-dimensional linear subspace.

Fig. 2. Example images of ten subjects of the Yale faces database taken under ambient lighting conditions.

The problem of clustering faces can then be interpreted as identifying the subspaces corresponding to each individual.

Following the established procedure of [**?**] we first use a global PCA to reduce the dimensionality of the data to $P = 5$ for GPCA and $P = 20$ for all other methods and perform clustering using standard subsets of the dataset of a varying number of clusters, from 2 to 10 [**?**]. We again compare PSC to GPCA, SCC, SSC and SLBF. We use PSC without pre-specifying K and R_k, but for all the competing methods all these parameters are fixed to be the true value of K, and the dimensionality of each subspace is set as $R = 2$.

Table I compares the mean clustering error and running time for each algorithm for the settings $K = 2, \ldots, 10$. It can be seen that PSC achieves perfect clustering accuracy with all subsets in less time than the other state-of-the-art methods. Furthermore, for all values of K, PSC was able to correctly determine the true number of clusters using the PRESS.

V. Conclusion

In this work we have introduced a novel and complete approach to subspace clustering which includes efficient model selection and detection of influential observations within the PCA framework. With simulations we illustrated that PSC performs well in a series of challenging situations which highlight the limitations of current approaches to subspace clustering. We have also shown that PSC achieves 100% accuracy on the Yale faces B database.

A number of important details about the PSC method have been omitted due to space constraints and will appear in forthcoming work. We have developed an upper bound of order $O(\sqrt{(\log N)/N})$ on the approximation error induced by the PAST approximation on the SVD. Further experiments with the Yale faces database have shown that our predictive influence function is able to identify influential observations with greater accuracy than the residual error and is particularly effective even when the number of training examples is small ($P > N$). We have also obtained a proof of convergence to a local optimal solution. Finally, the results presented here together with additional experiments on both real and simulated datasets show PSC is competitive with state-of-the-art methods whilst being computationally cheaper and with the ability to perform automatic model selection.

References

[1] R. Vidal, "Subspace Clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[2] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[3] J. Baek, G. McLachlan, and L. Flack, "Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1298–1309, 2009.

[4] P. Bradley and O. Mangasarian, "k-Plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, 2000.

[5] D. Wang, C. Ding, and T. Li, "K-Subspace Clustering," in *ECML PKDD*. Springer, 2009, pp. 506–521.

[6] U. Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, Aug. 2007.

[7] Yi Ma, "Generalized Principal Component Analysis: Modeling & Segmentation of Multivariate Mixed Data," 2006.

[8] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2009, pp. 2790–2797.

[9] G. Chen and G. Lerman, "Spectral Curvature Clustering (SCC)," *Int. J. Comput. Vision*, vol. 81, no. 3, pp. 317–330, Dec. 2008.

[10] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Arxiv preprint arXiv:1010.3460*, 2010.

[11] D. Belsley and E. Kuh, *Regression diagnostics: Identifying influential data and sources of collinearity*, 1st ed. New York, New York, USA: Wiley, 2004.

[12] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. New York: Springer-Verlag, 2002.

[13] B. Yang, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms," *Signal process.*, vol. 50, no. 1-2, pp. 123–136, 1996.

TABLE I
PERCENTAGE CLUSTERING ERROR AND COMPUTATIONAL TIME FOR SIMULATED DATA AND THE YALE FACES B DATASET

		Simulated data: the five scenarios					Yale faces dataset: the number of clusters, K								
		(a)	(b)	(c)	(d)	(e)	2	3	4	5	6	4	8	9	10
GPCA	e%	6.86	36.09	7.14	27.92	-	0.0	49.5	0.0	26.6	9.9	25.2	28.5	30.6	19.8
	time	2.07	3.26	1.40	7.22	-	1.42	2.72	4.91	8.08	11.71	33.11	99.49	286.36	1122.50
SCC	e%	0.00	0.00	0.00	29.49	33.05	0.0	0.0	0.0	1.1	2.7	2.1	2.2	5.7	6.6
	time	0.49	0.51	0.53	0.80	3.76	0.57	0.92	1.45	2.79	2.27	4.57	6.58	10.29	7.51
SSC	e%	0.0	0.0	11.83	38.33	12.64	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.4	4.6
	time	67.05	64.46	63.88	100.12	471.62	36.56	56.21	80.87	107.82	137.83	174.81	219.22	276.81	570.57
SLBF	e%	0.00	0.00	0.00	44.70	15.28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.9
	time	4.28	5.38	5.60	11.48	104.52	3.70	7.90	14.00	28.32	43.50	63.79	118.99	179.70	249.42
PSC	e%	0.00	0.03	0.12	4.09	0.78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	time	0.43	0.46	0.58	1.17	24.23	2.47	2.55	2.68	2.81	5.37	5.92	17.10	19.45	23.75