An overview of incremental feature extraction methods based on linear subspaces

Katerine Diaz-Chito[a,*], Francesc J. Ferri[b], Aura Hernández-Sabaté[a]

[a] *Centre de Visió per Computador, Universitat Autònoma de Barcelona, Spain*
[b] *Dept. d'Informàtica, Universitat de València, Spain*

Abstract

With the massive explosion of machine learning in our day-to-day life, incremental and adaptive learning has become a major topic, crucial to keep up-to-date and improve classification models and their corresponding feature extraction processes. This paper presents a categorized overview of incremental feature extraction based on linear subspace methods which aim at incorporating new information to the already acquired knowledge without accessing previous data. Specifically, this paper focuses on those linear dimensionality reduction methods with orthogonal matrix constraints based on global loss function, due to the extensive use of their batch approaches versus other linear alternatives. Thus, we cover the approaches derived from Principal Components Analysis, Linear Discriminative Analysis and Discriminative Common Vector methods. For each basic method, its incremental approaches are differentiated according to the subspace model and matrix decomposition involved in the updating process. Besides this categorization, several updating strategies are distinguished according to the amount of data used to update and to the fact of considering a static or dynamic number of classes. Moreover, the specific role of the size/dimension ratio in each method is considered. Finally, computational complexity, experimental setup and the accuracy rates according to published results are compiled and analyzed, and an empirical evaluation is done to compare the best approach

[*] Corresponding author

Email addresses: `kdiaz@cvc.uab.es` (Katerine Diaz-Chito), `francesc.ferri@uv.es` (Francesc J. Ferri), `aura@cvc.uab.cat` (Aura Hernández-Sabaté)

Preprint submitted to Knowledge-Based Systems *January 22, 2018*

of each kind.

Keywords: Incremental learning, Incremental feature extraction, Dimensionality reduction, Linear subspace methods, Covariance-based, Covariance-free

1. Introduction

Processing large amounts of data is nowadays a challenging task in the field of pattern recognition, which aims to extract meaningful information embedded in the data. As a first general step, appropriate structured descriptors or features must be selected or extracted from raw data through a learning process using any prior information available. This leads to a more discriminative data representation with lower dimensionality, facilitating the following steps on machine learning and data mining pipelines. The traditional way to extract these features is usually based on batch learning. However, this requires that all the data must be available from the beginning and used as whole, which is not convenient or even feasible in most online, interactive or stream-based processing applications. Several application domains such as autonomous navigation systems [1], human-robot interaction [2], object tracking [3], image classification [4], stream processing [5], face recognition [6] or recommendation systems [7, 8, 9, 10] have been shown as examples where a complete set of training samples is usually not known in advance but generally provided little by little. Moreover, in some cases the properties of data may change as new data is considered. For instance, in face recognition tasks, human faces may show large variations depending on expressions, lighting conditions, make-up, hairstyles, aging and so forth. When a human is registered in a person identification system, it is quite difficult to consider all this facial variability in advance [11] but instead it is more convenient to discover it during the operation of the system.

As an effective alternative, the paradigm of incremental or adaptive learning has been considered and deeply studied as its own pattern recognition and machine learning subfield. By using incremental learning, feature extraction processes should be capable of incorporating the new information available while retaining the previously acquired knowledge, without accessing the previously processed training data. This fact is very challenging specially in the era of big data, where new chunks of data is continuously appearing and new classification objectives arise.

Among the huge amount of incremental learning schemes, this paper focuses on linear subspace-based incremental feature extraction methods with orthogonal matrix constraints based on global loss function, due to the extensive use of their batch approaches versus other linear alternatives with unconstrained objectives, such as probabilistic PCA [3, 12], or matrix factorization methods [7, 13, 8, 14, 9, 10, 15, 16], mostly popular for building collaborative filtering on recommender systems. Note that not all linear feature extraction methods need to produce orthogonal projections, or indeed projections at all. While subspace-based methods can be based on linear and non-linear subspaces, linear methods are the most extensively used, even in highly non-linear problems where the non-linearity is modeled in the subsequent feature extraction and classification stages instead. An example of this is the use of linear dimensionality reduction methods in modern deep learning architectures as preprocessing step to reduce the number of parameters to be learned and the number training samples [17, 18, 19]. Moreover, these techniques have been used in the last years in many successful problems as object tracking [20, 21, 22] or in other application fields, such as pharmaceutics [23], medical image [24, 25], agriculture [26], industrial applications [27], chemometrics [28, 29] pattern recognition [30] or bioinformatics [31, 32].

Therefore, this paper presents a categorized overview of the research done over the past decades on linear subspace-based incremental feature extraction and dimensionality reduction for matrices and general applications. Special emphasis is put on those methods with orthogonal matrix constraints based on global loss function, such as Principal Components Analysis (PCA), Linear Discriminative Analysis (LDA), and Discriminative Common Vector (DCV) methods, over methods with unconstrained objectives, such as probabilistic PCA

[3, 12] or matrix factorizations [7, 8, 14, 9, 10, 15, 16]. Similarly, we consider that those incremental methods which are more related to subspace-to-subspace matching [33, 34], and tensor factorization [35, 36, 37, 38] are out of the scope. By restricting ourselves to these methods, we can both keep our survey to a manageable size and also concentrate at the basic ideas behind the different incremental approaches that are usually shared across a wider range of works. For the same reason, we have obviated incremental nonlinear extensions of the above methods [39].

In the present work we will differentiate methods according to the **subspace model** used. From this viewpoint, two main categories of incremental subspace-based methods are usually considered depending on whether or not the above matrices are explicitly considered and computed (using different forms of decompositions) or not. Some of these variants are referred to in the literature as covariance-based or covariance-free methods. Table 1 summarizes all the papers considered in the present work according to the subspace model used and the computation (or not) of the above matrices.

To complete this multidimensional taxonomy, which is graphically illustrated in Figure 1, different ways of feeding incremental algorithms are considered. The first one is in terms of the data size required for each update, which may range from one single sample at a time to moderate chunks of data. The second one is in terms of data labels, i.e. whether or not the set of labels in the corresponding classification problem is fixed beforehand or may grow arbitrarily along the incremental process. We will refer to these two aspects as **chunk size** and **chunk label** structure, respectively. Finally, we will also consider the **size/dimension** ratio, where we explicitly distinguish between the case in which the input space dimension is much greater than the expected data stream size and this constitutes a requirement or strongly conditions a particular method. Facing very small values of this ratio is usually referred as the small sample size (SSS) case.

The paper is organized around the above taxonomy paired with the discussion of the advantages and disadvantages inherent to each approach. The

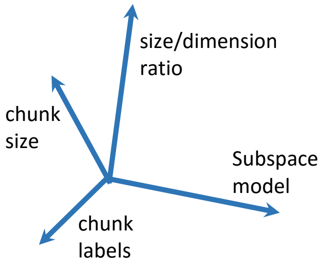


Figure 1: Proposed taxonomy for subspace-based incremental feature extraction methods.

remainder of the paper is structured as follows. Section 2 describes the problem setting. Sections 3, 4 and 5 contain an organized overview of incremental feature extraction based on PCA, LDA and DCV approaches, respectively. Section 6 shows a performance analysis of the incremental methods regarding their

Table 1: Summary of the methods presented in the paper.

		System updating way			
		covariance-based	using SVD updates	adaptive	covariance free
Approach based on	PCA	Murakami [40] Hall [43, 44, 45] Ozawa [47, 11, 48] Li [51] Huang [54] Duan [56] Arora [57] Jin [58]	Chandrasekaran [41] Levy [12] Kwok [49] Zhao [52] Li [55]		Weng [42] Skočaj [46] Qu [50] Yan [53] Zeng [5]
	LDA	Pang [59, 60] Ye [62] Kim [64, 65] Uray [68] Song [70] Zheng [72] Lamba [74] Peng [6] Lu [75] Dhamecha [76]		Zhao [61] Liu [63] Lu [66, 67] Yeh [69] Chu [71] Zhang [73]	
	DCV	Diaz [77, 78] [4, 81]			Ferri [79, 80] Diaz [77] Lu [82] Zhu [83]

experimental setup and accuracy rates available in published results, as well as the empirical comparison of the best approach of each kind. Finally, Section 7 summarizes the main contributions of this survey.

2. Problem Setting

Throughout this paper, the d-dimensional vector space, $\mathbb{R}^d$, will be considered as the input representation space. $U_r = [u_1 \dots u_r]$ represents a projection mapping, where u_i are orthonormal vectors that span a linear subspace in $\mathbb{R}^d$ and usually $r \ll d$. Given a (column) vector, $x \in \mathbb{R}^d$, its mapping onto $\mathbb{R}^r$ is $U_r^T x$, and the corresponding orthogonal projection onto the subspace defined by U_r is given by $U_r U_r^T x$. Different methods optimize different criteria which are based on different scatter matrices [1] defined from training data in order to obtain an appropriate mapping, U_r.

For example, given the unlabelled data matrix $X = [x_1 \dots x_m] \in \mathbb{R}^{d \times m}$ and its centered version, X_c, the total scatter matrix, $S_t = X_c {X_c}^T$, is a positive semidefinite matrix which is just a scaled version of well-known sample estimate of the covariance matrix. If labels in the X matrix are known, the total scatter matrix can be decomposed in two parts, $S_t = S_w + S_b$. The within-class scatter matrix, $S_w = X_{cc} {X_{cc}}^T$, is an averaged representation of the dispersion of data with regard to corresponding class averages [2]. The between-class scatter matrix, $S_b = \overline{X}_c {\overline{X}_c}^T$, represents the dispersion of class means with regard to the overall mean, where $\overline{X}_c$ is a matrix that contains class centroids (centered with regard to the overall mean).

The eigendecomposition or eigen-value/vector decomposition (EVD) of S_t

[1] Scatter, covariance or correlation matrices correspond to different but very closely related concepts in this context. Thus, we will refer to scatter matrices indifferently.

[2] We use here subindices c and cc to express centering with regard to the overall mean or to the class mean, respectively. This distinction will not be used if the meaning is clear from the context.

can be written in general as

$$X_c{X_c}^T = U\Lambda U^T = [U_r \;\; U_0] \begin{bmatrix} \Lambda_r & \\ & 0 \end{bmatrix} \begin{bmatrix} {U_r}^T \\ {U_0}^T \end{bmatrix},$$

where $U = [u_1 \ldots u_d]$ is a column matrix formed by the eigenvectors associated to the eigenvalues, $\lambda_1 \geq \ldots \geq \lambda_d$, in the diagonal matrix Λ, and $\lambda_i = 0$ for all $i > r$. Note that U_r and U_0 are bases of two complementary subspaces, the *range*, $\mathcal{R}(S_t)$, and *null*, $\mathcal{N}(S_t)$, subspaces, respectively. Applications require dealing with subspace models, U_r, that approximate $\mathcal{R}(S_t)$ with reduced sizes. When new training data, $Y \in \mathbb{R}^{d\times n}$, $1 \leq n \ll m$ becomes available, the effective new training set is $\widetilde{X} = [X \;\; Y]$. The challenge then is to obtain the feature extraction model, $\widetilde{U_r}$, associated to $\widetilde{X}$ by accessing and processing only Y and the current model, U_r.

3. Incremental Principal Components Analysis (IPCA)

PCA is the most popular feature extraction method which aims to find a set of orthonormal basis vectors (the principal components) that maximize the variance over all the data when it is projected onto the subspace spanned by these principal components. Formally, it corresponds to the following optimization problem

$$\max |U^T S_t U| \quad \text{s.t. } UU^T = I$$

Although PCA is not optimal with regard to discrimination since no class information is used to obtain principal components, it is optimal in terms of minimum reconstruction error. Thus, PCA and closely related methods are very widely used in a large range of domains both in batch and incremental versions.

It can be shown that the above optimal solution is obtained through the EVD of S_t:

$$S_t = X_c{X_c}^T = [U_r \;\; U_{\overline{r}}]\ [\Lambda_r \;\; \Lambda_{\overline{r}}]\ [U_r \;\; U_{\overline{r}}]^T,$$

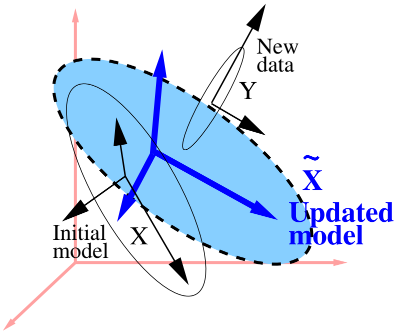


Figure 2: Subspaces and corresponding models involved in IPCA methods. The initial training set, X, and the update set, Y, are represented using thin solid lines and the resulting updated set is represented using thick dashed lines and shades.

for a given dimensionality, r, where the mapping U_r maximizes the variance of the projected data and $U_{\overline{r}}$ corresponds to its complementary subspace that involves the smallest nonzero eigenvalues. Note that the whole $\mathcal{R}(S_t)$ is represented by $span\{[U_r \;\; U_{\overline{r}}]\}$.

As only the restricted model (U_r, Λ_r) is usually kept for large problems, when a given data set, X, is augmented with new data, Y (which may carry out its own subspace model), then the problem is to update U_r to a new model $\widetilde{U_r}$ which maximizes the variance when projecting $\widetilde{X}$. This situation is schematically shown in Figure 2 in which both the initial model corresponding to data X and the new data, Y are represented as ellipsoids with arrows corresponding to principal directions. The updated model corresponding to $\widetilde{X} = [XY]$ is represented in the same way but with color shade and dashed lines.

Since an exact EVD update at each iteration in IPCA is not possible and not feasible in terms of space complexity as the *range* subspace dimension could grow with new data, the different IPCA methods differ in the way the intermediate computations and approximations are done in order to update U_r in an efficient and convenient way.

Instead of the EVD, some IPCA methods use the singular value decomposition (SVD) as an alternative to obtain a convenient decomposition of S_t. Other methods use SVD as an orthogonalization tool or conveniently update approximations of the SVD. Several early works [84, 85, 86, 87] analyzed numerical issues related to both EVD and SVD updates including sensitivity of the corresponding subspace models. These works lead to the first proposals in which PCA is explicitly formulated in an incremental way using particular subspace models which are based either on the EVD of a covariance-like matrix or the SVD of the data matrix.

In the following subsections and for the sake of readability, we split the methods according to the way they update the system. In subsection 3.1, we detail covariance-based methods, whose aim is to maintain and update a more or less explicit model of the scatter matrix using mainly EVD. In subsection 3.2, we summarize the methods based on partial SVD updates that modify principal components without constructing or referring to a covariance-like matrix. Finally, in subsection 3.3 we review purely adaptive methods which are usually referred to as covariance-free.

3.1. Covariance-based Incremental PCA

In 1982, Murakami et al. [40] first posed a formal version of IPCA applied to image analysis. This method updates the model with exactly one new sample at a time, by solving a small eigenproblem at each iteration and assuming no mean recomputation.

Properly updating the mean is of utmost importance in many practical applications because it is a crucial representative characteristic of a cluster of observations and it severely affects the obtained mapping and corresponding subspace. In 1998, Hall et al. [43] proposed an IPCA with uncentered data. In this case, the mean is recomputed for each update given the previous mean value and the new sample, and this is taken into account in the corresponding decomposition update. In particular, for each new sample $x \in \mathbb{R}^d$, the current mapping U_r needs to be updated by adding a new orthonormal vector which is the normalized residue of the new centered sample, $x_c = x - \overline{x}$, which is nothing but the orthogonal projection of x_c onto the complementary subspace of U_r, which is given by $(I - U_r U_r^T)x_c$, where I represents the identity matrix. This updated mapping is only one rotation away from the new decomposition.

A major problem comes from discarding nonzero eigenvalues along the iterative process. Three possible criteria for keeping eigenvectors and eigenvalues are considered in [43] using either (1) an absolute threshold, (2) the largest eigenvectors and (3) the eigenspectrum energy. The same authors developed in [44, 45] an enhanced and extended version of their previous approach which is referred to as Merging and Splitting EigenSpaces (MSES) because it serves both to update and downdate previous models. In this generalized formulation, a group (chunk) of samples, $Y \in \mathbb{R}^{d\times n}$, is considered for each update. The mean is also incrementally recalculated and different subspace updates are considered. As a main result, it is shown that the final scatter matrix after the update can be written as the sum of three similar matrices: the one corresponding to initial data, X, a low-rank matrix corresponding to new data, Y, and a third rank-one matrix that involves the means of both sets of samples. This or other very related decompositions have been used in most incremental feature extraction methods using linear subspaces. The MSES method then takes into account the EVDs of old and new data together with their means. The (few) eigenvectors corresponding to new data are projected using $(I - U_r U_r^T)$ in order to obtain a residue matrix corresponding to the chunk, which still needs orthonormalization prior to its addition to the current principal components.

Ozawa et al. [11, 47] devised a one-sample version of the MSES approach [44, 45] in which the model is updated in such a way that an eigenaxis is augmented based on an accumulation ratio of the initial and new eigenvalues. This strategy helps in controlling the dimensionality of the eigenspace. Li presented in [51] a reformulation of the incremental PCA that weighs each new sample in the target model as $\alpha U_r \Lambda_r U_r^T + (1-\alpha)xx^T$ and whose update rule reduces to the eigendecomposition of a small matrix. This approach can be further extended to account for outliers which leads to a robust incremental algorithm (RIPCA). These approaches assumed that there is no change to the mean during covariance updating, and an exact mean update is not provided. In addition, the update rate is decided empirically in order to adjust the rate between the current samples and the newly observed sample in combining the new mean vector.

In fact, it is very difficult to fulfill the two roles at the same time [56].

In [48], Ozawa et al. extended their previous approach [11, 47] to a Chunk-IPCA. The eigendecomposition of the covariance matrices is the same as in Hall et al. [45], but with an updated accumulation ratio of the initial and new eigenvalues, in order to control the dimensionality of the eigenspace. In most cases, there are no significant differences in performance compared to the use of batch learning, but, like in the one-sample update case, the threshold value has to be optimized for each dataset.

Huang et al. [54] proposed a mean-shifting PCA and its incremental method based on the autocorrelation matrix. The incremental version does not involve increasing the matrix dimension with the number of input data. This approach produces the same eigenspaces as the MSES approach [44] up to possible numerical errors in the different processes.

G. Duan et al. [56] came up with the chunk version of the RIPCA approach [51]. While in Li's algorithm the mean update is not considered, in this version both the mean and the covariance matrix are updated. Furthermore, a weighting matrix is used instead of a single weight to balance the contribution between the previous data and the new observed data towards the new subspace.

Arora et al. [57] formulated an IPCA approach as Stochastic Approximation (SA) problem based on the incremental SVD of Brand [88] but using EVD. The samples are processed one at a time, resulting in a rank-one update to the unnormalized update matrix.

Jin et al. [58] elaborated an incremental/ decremental PCA framework based on an EVD updating and downdating algorithm, referred to as EVD Dualdating (EVDD). This algorithm permits simultaneous arbitrary addition and deletion operations, by transforming the EVD of the covariance matrix into a SVD updating problem.

In general, the projection matrix is updated by extending the initial submatrix with a new base that spans to the new subspace. Many approaches had been proposed to control the updating by normalizing the eigenvalues [44], or by applying a ratio [11, 51, 48].

3.2. Incremental PCA using SVD updates

In 1997, S. Chandrasekaran et al. [41] suggested an updating algorithm which adds a raw input vector directly, followed by the standard SVD algorithm. The new sample is projected onto the initial complementary subspace and normalized, and then a new eigenproblem is formulated and solved by using SVD. As in most early approaches, the mean vector is not updated.

Levy et al. [12] introduced the Sequential Karhunen-Loeve (SKL) approach in which the update is reduced to a modified SVD algorithm [89] which is efficient in the SSS case.

Kwok et al. [49] evolved a chunk incremental PCA approach via the SVD updating algorithm [90], known as SVDU-IPCA. It relies on the autocorrelation matrix, and it does not allow updating the mean. It suffers from the problem of growing demand for storage and computation, because the size of the autocorrelation matrix increases with the new data, and an additional process is needed to transfer the resulting right singular vectors and kept whole data to principal components. This work was rewritten by Zhao et al. [52], providing more comparisons and experiments.

Li et al. [55] explored an incremental version of the batch updating/ downdating algorithm of SVD proposed by Brand in [91] called Accurate-IPCA (AIPCA), which provide both update and downdate samples based on the matrix additive modification presented by Hongyuan et al. in [92].

3.3. Covariance-free Incremental PCA

The Candid Covariance-free IPCA (CCIPCA) was introduced by J. Weng et al. in [42] to incrementally compute the principal components of a sequence of samples, keeping the scale of observations. This method is motivated by the concept of statistical efficiency, so the estimate has the smallest variance given the observed data. An amnesic mean technique is also used to dynamically determine the retaining rate of the initial and new data, instead of a fixed learning rate. The CCIPCA algorithm generates observations in a complementary space for the computation of the higher order principal components. However, an approximate centric alignment on the input data is applied in this complementary space, where only the current sample is correctly centered.

In [46], Skočaj et al. performed an incremental algorithm for building face representations using reconstructive information whose subspace model is updated with a single new sample at a time. This approach keeps all low-dimensional coefficients and reduces the update to computing the (batch) PCA in a reduced dimension. Qu et al. [50] improved the subspace updating strategy on this method by means of the concept of adaptive subspace to adjust subspace updating. This consists of using two thresholds to differentiate if the sample has to be added to current model or not, such that the subspace dimension do not increase rapidly, and the computational cost and storage are saved regarding the batch method.

Largest-Eigenvalue-Theory based Incremental Principal Component Analysis (LET-IPCA), was presented by Yan et al. [53]. LET-IPCA is based in the well-known power method [89] and achieves the estimations of the leading eigenvectors by cooperatively and iteratively preserving the dominating information. Similarly to some previous approaches, the mean is not updated.

Zeng et al. [5] proposed a version of CCIPCA [42], with exact historical mean update, called Centered Incremental Principal Component Analysis (CIPCA), where not only the current sample is centered like in CCIPCA, but also all historical data are correctly updated by the current mean. Moreover, CIPCA only needs to keep the learned eigenvectors, and several first-order statistics from the past samples, such as the mean and the number of samples. CCIPCA also converges more quickly, and the performance improvement is especially obvious when the data's inherent covariance is not stable.

Most covariance-free IPCA approaches start by projecting the new data (or a representation of this) on the complementary subspace of the previous data. Then, an orthogonal base is calculated and a new system is formulated. The projection matrix is finally updated by extending the previous one with a new orthogonal base resulting from a normalization process.

Table 2: Overview of the Incremental PCA algorithms

Author	Year	Acronym	mean update	Subspace model	Chunk size-lab	size/dim ratio	Application
Levy [12]	2000	SKL		SVD	∴	$\ll$	Image classification
Hall [44, 45]	2000/2	MSES	✓	EVD	∴ ∵		Face recognition
Skoč [46]	2003	IPCA-REC	✓	EVD	●		Visual learning
Weng [42]	2003	CCIPCA		Gradient rule	●		Image analysis
Kwok [49], Zhao [52]	2003/6	SVDU-IPCA		SVD	∴	$<$	Face recognition
Ozawa [47, 11]	2004/6	MMSES		EVD	●		Pattern classification
Li [51]	2004	RIPCA		EVD	●	$<$	Background modelling
Yan [53]	2005	LET-IPCA		Power method	●	$<$	Image classification
Ozawa [48]	2008	MMSES-C	✓	EVD	∴		Pattern classification
Huang [54]	2009	MS-IPCA	✓	EVD	∴	$<$	Image recognition
Duan [56]	2011	RIPCA-C	✓	EVD	∴	$<$	Face recognition
Li [55]	2011	AIPCA		SVD	∴ ∵	$\ll$	Dimensionality reduction
Arora [57]	2012	SA-IPCA		EVD	●	$<$	Image analysis
Zeng [5]	2013	CIPCA	✓	Gradient rule	●		Streaming data
Jin [58]	2014	EVDD	✓	EVD	∴ ∵	$\ll$	Image Classification

3.4. Discussion

Table 2 summarizes in chronological order the collection of the IPCA algorithms presented herein. Each algorithm is identified by the name of the first author in the main publication presenting the algorithm and an *acronym* for the method itself. Then, we indicate whether an exact mean update is considered or not, followed by the type of *Subspace model* in which the algorithm is mainly based, or the kind of adaptive rule in case of covariance-free methods. The next columns in the table correspond to *chunk size-lab* and *size/dimensionality* ratio, respectively. Chunk-based methods are marked as ∴ and sample-based ones as ●. The symbol ∴ ∵ is used to denote that the method allows decremental updates. The symbols $<$ or $\ll$ indicate to which extent the SSS assumption is a requirement or not for the particular algorithm proposal. The last column in the table shows the main application domains where the approach has been validated.

Since the space and the computational complexity are the most important indicators to evaluate the algorithm efficiency, we perform a comparative study among the different cited algorithms in terms of the main decompositions employed and the computational complexity. Table 3 shows a summary of the main steps in each algorithm along with the computational complexity, organized according to chunk size. In each part, methods are ordered chronologically. Covariance-based and methods using SVD updates are shaded with dark and

Table 3: Main decompositions and computation complexity on IPCA approaches. Methods are ordered chronologically, showing first the methods with correction per sample and then the chunk-based ones. Covariance-based and methods using SVD updates are shaded with dark and light gray, respectively.

Approach	Update	Decomposition(dim)	Computational complexity
MSES [43]	●	Norm(d)+EVD($r+1$)	$O((r+1)^3)$
IPCA-REC [46]	●	Norm(d)+EVD(r+1)	$O(dr+(r+1)^3)$
CCIPCA [42]	●	Norm(d)	$O(dr)$
MMSES [47]	●	Norm(d)+EVD($r+1$)	$O((r+1)^3)$
RIPCA [51]	●	EVD($r+1$)	$O((r+1)^3)$
LET-IPCA [53]	●	Norm(d)	$O(dr)$
SA-IPCA [57]	●	EVD($r+1$)	$O((r+1)^3+dr^2)$
CIPCA [5]	●	Norm(d)	$O(dr)$
SKL [12]	∴	QR(d,$r+n$)+SVD($r+n$)	$O(d(r+n)^2)$
MSES [45]	∴	EVD(d,n)+GSO(d, r_y+1)+EVD(d, $\tilde{r}$)	$O(\tilde{r}^3)$
SVDU-IPCA[52]	∴	QR(d, n)+SVD($r+r_y$)	$O(n^3+(r+n-\tilde{r})n^2+(\tilde{r}+n)^2\tilde{r})$
MMSES-C [48]	∴	EVD($\tilde{r}$)	$O(\tilde{r}^3+d(r+r_y)r_yn)$
MS-IPCA [54]	∴	EVD($r+n+1$)	$O((r+n+1)^3+d(r+n+1)^2)$
RIPCA-C [56]	∴	EVD($r+n+1$)	$O((r+n+1)^3)$
AIPCA [55]	∴	QR(d,n)+SVD($r+n$)	$O(d(r+n)^2+(r+n)^3)$
EVDD [58]	∴	QR(d,n)+SVD($r+n$)+EVD($r+r_y$)	$O(dr(r+n))$

light grey, respectively. Second column describes the kind of updates applied. The third column shows the decomposition used for each method in terms of its dimensionality, to give an idea of the space complexity. The last column shows the computational complexity of the method described in the original article. As notation, EVD(dim) is used to indicate a standard eigendecomposition of size dim, and SVD(dim), QR(dim) and GSO(dim) to refer to (thin) singular value decomposition, QR decomposition, and Gram-Schmidt orthogonalization of dim size, respectively. Normalization and other straighforward vector operations of size dim are represented as Norm(dim). Other important matrix operations that may strongly conditionate the cost in some cases are not explicitly indicated in the table for clarity reasons. The different (preserved) ranks of old, new and resulting data are marked with r, r_y, and $\tilde{r}$, respectively, with $r_y < r$ and $\tilde{r} \leq (r + r_y)$. In case the original article does not provide any information about the computational complexity, we report, in gray letters, the asymptotic cost expressions corresponding to standard implementations of well-known operations and decompositions.

Several conclusions emerge from these tables. IPCA covariance-based algorithms using one sample updates, present the same complexity, $O(r^3)$, since all of them update the projection matrix by extending it with a normalized residue vector, leading to a new eigenproblem in the r-dimensional space to obtain the resulting eigenvectors. The main difference among them is the threshold criterion to decide whether the new sample is incorporated into the training set.

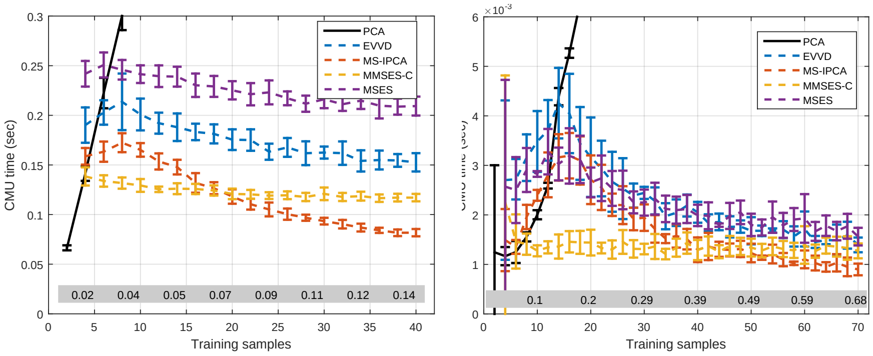
The complexity of the IPCA covariance-free methods using one sample update is dominated by $O(dr)$, since they require only a normalization process and several straightforward vector-matrix operations which leads to very efficient algorithms compared to the previous ones, provided that $r > \sqrt{d}$. The price to pay is the approximative nature of these methods and the potential convergence problems. In fact, LET-IPCA [53] can converge much faster than CCIPCA which relies on the assumption that previous eigenvectors are well estimated. Notice that none of the IPCA covariance-free methods has been extended to deal with chunks of data. The complexity of chunk-based IPCA algorithms involve higher complexity and variability than the previous cases. In particular, all methods are at least cubic in their main parameters.

An empirical validation has been performed to show the complexity of some of these methods. Two cases have been considered, $d >> m$ and $d > m$. Table 4 shows the main characteristics of the datasets used to validate the IPCA approaches.

Table 4: Datasets used in IPCA validation along with their corresponding details. c is the number of classes. m_j is the number of samples per class.

	size	c	m_j	Variability type
CMU-PIE [93]	110×150	68	56	Faces (pose & light)
NIST [94]	32×32	10	100	Handwritten digits

In all experiments, an initial model with $m_j = 2$ is obtained (using the corresponding batch algorithm) and then 2 samples per class are progressively added to the previous learnt model until the 70% of the dataset has been used. The remaining 30% is used as test set. Cross validation is applied as evaluation protocol to avoid bias to a particular training/testing split. Each experiment is run 10 times with different random training/testing sample choices. As classifier, a simple 1-Nearest Neighbors is employed, using the Euclidean distance. All algorithms have been run on a computer with an Intel(R) Core(TM) i7-4790



(a) CMU-PIE

(b) NIST

Figure 3: Comparison among the computational training time.

CPU @ 3.60GHz, 3601 MHz, and 32-GB RAM. Figure 3 shows the average of the computational training time over the iterations as well as the corresponding dispersion bars. The greyscale bar in the figure represents the ratio between the number of samples in training set and the dimension of the original space. The differences among the methods are more evident in the $d >> m$ case that when $d > m$.

Regarding the classification performance, Table 5 shows the accuracy rate in the last iteration.

From the results shown in Table 5 and Figure 3, we observe that, among the validated methods, MS-IPCA [54] shows good behavior regarding performance and cost in both cases, $d >> m$ and $d > m$.

Table 5: Performance comparison on the last iteration.

	CMU-PIE [93]	NIST [94]
PCA	79.1±1.0	99.5±0.3
EVVD [58]	72.3±1.2	96.4±1.1
MS-IPCA [54]	**73.0±1.2**	**97.1±0.9**
MMSES-C [48]	41.0±8.6	94.5±2.5
MSES [45]	72.3±1.2	95.8±1.6

4. Incremental Linear Discriminant Analysis (ILDA)

LDA is a traditional statistical technique that reduces dimensionality while preserving as much of the class discriminatory information as possible, Instead of maximizing variability, LDA tries to find appropriate subspaces in which labelled data gets optimally separated. To this end, the total scatter matrix can be decomposed in two parts, $S_t = S_w + S_b$, such that one wants minimal within-class dispersion and maximal between-class dispersion at the same time. The classical way to achieve this, consists of repeating the rationale behind PCA but using the ratio $S_w^{-1} S_b$ instead of the total scatter matrix, S_t. That is,

$$\max \frac{|U^T S_b U|}{|U^T S_w U|} \quad \text{s. t. } UU^T = I$$

This way of formulating the problem brings about two very important consequences: 1) the number of discriminant dimensions is bounded by $(c - 1)$, which is the rank of S_b, with c the number of classes in the training set, and 2) the matrix S_w must be nonsingular which is a very critical point in the SSS case [95] in which the size/dimension ratio is lower than one.

To increase its applicability, many LDA extensions, such as Fisherfaces [96], direct LDA [97], null space based LDA [98], complete LDA [99], LDA/QR [100] or LDA/GSVD [101], have been developed in the last decades. These extensions try to maintain the same rationale and overcome singularity problems either by first projecting the problem in a convenient subspace, through regularization, or using alternative indirect or approximate optimizations.

As in the case of PCA, many ILDA approaches have been introduced to deal with large problems in which discriminant components need to be calculated and updated in a sequential way. Figure 4 illustrates the subspaces involved when updating LDA models with several labels. In this case, the initial data, X, has two different classes which are represented as two ellipses labelled as 1 and 2 and the shown arrow corresponds to the most discriminating direction. Exactly as with the new coming data, Y which contains three different labels. The

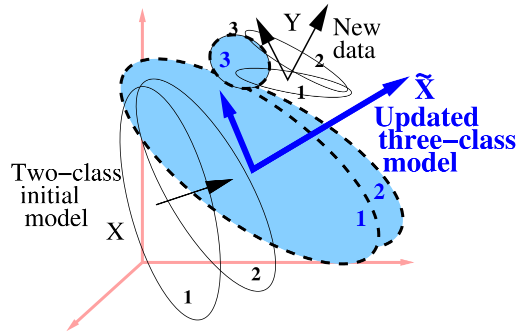


Figure 4: Subspaces and corresponding models involved in ILDA methods. X represents the initial training set with two classes (1 and 2). Y is the update set with new samples for existing classes (1 and 2), as well as a new unknown class (3). $\tilde{X}$ represents the resulting updated set. Data corresponding to initial and update sets are represented using thin solid lines and the resulting data is represented using thick dashed lines and shades.

problem now corresponds to obtaining the updated discriminating directions corresponding to the whole data, $\tilde{X}$, which is represented as shaded ellipses and dashed lines.

The variability of different ILDA methods is even higher than the one for IPCA, but nevertheless we distinguish only between covariance-based ILDA approaches and other adaptive ILDA methods. The first kind of approaches, detailed in subsection 4.1, try to get close to the classic LDA rationale and are based on an EVD of an appropriate matrix. The second type of methods are summarized in subsection 4.2 and use either local learning rules or adapted SVD or QR decompositions to update subspace models.

4.1. Covariance-based Incremental LDA

Among the first incremental proposals we have the one from Pang et al. [59, 60], which we will refer to as Chunk ILDA (CILDA). When new labelled data is available, the current scatter matrices, S_w and S_b are updated to obtain $\widetilde{S}_w$ and $\widetilde{S_b}$. Then the eigenaxes of the new feature space are obtained by solving the eigenproblem associated to $(\widetilde{S}_w^{-1}\widetilde{S_b})$. If $\widetilde{S}_w$ is singular, the algorithm projects the samples in the range of $\widetilde{S}_w$. The method can manage chunks of different size containing new, previously unseen classes. The advantage of this algorithm is that recomputing $\widetilde{S}_w$ and $\widetilde{S_b}$ from scratch is not required. However, the complexity is very high because the EVD of the matrix $(\widetilde{S_w}^{-1}\widetilde{S_b})$ is computationally expensive (of the order of $O(d^3)$), and the chunk size can not be too large since the memory cost will become very expensive. In addition, this approach can not address the SSS problem.

Almost at the same time, Ye et al. [62] proposed another ILDA approach named Incremental Dimension Reduction Algorithm via QR Decomposition (IDR/QR). This algorithm updates the current model at each time step by incorporating a single new sample that may belong to a previously unseen class. IDR/QR applies QR decomposition to obtain the optimal projection matrix in the subspace spanned by the class means. Both S_b and S_w matrices are projected onto the lower-dimensional space of the class means, where regularized LDA is performed using EVD. IDR/QR algorithm is fast when the number of classes is low, since the dimension of the subspace is limited by the number of classes. Large number of classes would lead to high computational complexity. Moreover, the main disadvantage of this algorithm, as shown in [61, 71], is that large quantities of convenient and discriminant information are lost in the updating process.

Kim et al. [64] created an ILDA algorithm that appropriately solves the SSS problem, in which a chunk of new samples for known classes, as well as unknown class are considered. This approach is called IncLDA, and applies the concept of *Sufficient Spanning (SS)* set[3] to perform the update. At each updating step, the SS set is exploited to reduce the size of the scatter matrices yielding a speed up similar to MSES [44]. Nevertheless, one of the main drawbacks is the gap in performance with regard to batch LDA due to the approximations used [82]. Kim et al. enriched the IncLDA approach with a more thorough analysis, discussions and new experiments in [65].

Another incremental approach that solves the SSS problem was the ILDA augmented PCA (ILDAaPCA), stated by Uray et al. in [68], using a single or

[3] *A Sufficient Spanning* set (SS), Φ, is a reduced set of basis vectors spanning a subspace containing most of data variation, in the sense that the associate orthogonal projection, $\Phi\Phi^T X$, approximates the corresponding data matrix, X.

a chunk of new samples of either existing classes or previously unseen ones. ILDAaPCA combines reconstructive and discriminative information during training, where IPCA-REC [46] is used as reconstructive model generating an extended representation of the samples. Plain LDA is then applied on the new representation as discriminative model.

The Incremental Weighted Average Samples (IWAS) algorithm was formulated by Song et al. [70] for facial feature extraction tasks using single sample updates (from existing or new classes). This incremental algorithm, able to work under the SSS case, maps the input space into an intermediate subspace spanned by class centroids like the IDR/QR [62].

Zheng et al. advanced in [72] the Incremental Dual-Space LDA (IDSLDA). The corresponding batch method [102] is modified in order to reduce the computational complexity, and to incrementally update the discriminant vectors when new samples are inserted. The IDR/QR method is adopted using the class means once they are projected onto the *Difference Subspace*[4] (DS) of the training samples. Then, it finds a base to project the scatter matrices, and calculate the eigenvectors of $(\widetilde{S_w}^{-1}\widetilde{S_b})$. Moreover, IDSLDA has been extended to update the training set with either one sample from existing or unknown classes.

Lamba et al. [74] extend the IncLDA [64, 65] to Incremental Subclass Discriminant Analysis (ISDA), obtaining the incremental version of the Subclass Discriminant Analysis (SDA) [103]. They showed that when the underlying data from the same class conforms to multiple normal distributions, it is useful to consider each of them as a subclass.

Peng et al. [6] presented an extension of the IDR/QR approach [62] in which chunks can contain samples from existing and new classes simultaneously. The authors name this method as c-QR/IncLDA.

[4] The *Difference Subspace* (DS) of a class is defined as $B_j = span\{x_j^i - x_j^1\}$, where the first samples of each class are taken as the subtrahend vector. These subspaces are summed up to form the complete difference subspace as $B = B_1 + \ldots + B_c$. B and the *Range* space of S_w are the same space.

Another chunk-IDR/QR approach is presented by Lu et al. in [75] where approximation is not used, such that there is not a gap in performance between incremental and batch IDR/QR solutions.

Dhamecha et al. [76] presented an incremental semi-supervised discriminant analysis algorithm called ISSDA, which uses the unlabeled data for enabling incremental learning and the sufficient spanning set representation of scatter matrices of Kim et al. [64, 65]. This approach incrementally learns the between-class variability and uses unlabeled data to learn the overall variability. The eigenmodel of S_b is learnt from incremental batch and merged with the existing eigenmodel. The new discriminative components are obtained by using the updated eigenmodel of S_b and an offline estimated eigenmodel of S_t.

4.2. Adaptive Incremental LDA

Zhao et al. [61] presented an incremental algorithm, GSVD-ILDA, based on the SVD update. This approach considers chunks of samples of either known or unknown classes. The core step of GSVD-ILDA is to update the eigenvectors of the centered data matrix. GSVD-ILDA algorithm suffers from a common problem, the difficulty to determine to which degree the performance should be traded off for efficiency. If too many minor components are removed, the performance will deteriorate. Otherwise the efficiency will be low. Moreover, the performance is sensitive to parameter settings, while tuning the parameters is not an easy task [63].

Liu et al. [63] proposed the Least Square Incremental LDA (LS-ILDA) approach based on a previous batch approach [104], where Multivariate Linear Regression (MLR) is used for model construction along with a least squares criterion. It can be shown [104] that the projection matrix obtained through MLR is equivalent to the LDA one, in the sense that they represent equivalent subspaces. The LS-ILDA works differently depending on the sample size to dimensionality ratio. In the SSS case, only the pseudoinverse of the centered data matrix, X_c, needs to be updated. Otherwise, the method needs to manage and update the pseudoinverse of $X_c X_c^T$. Although LS-ILDA produces the same solution than plain LDA, the amount of information it needs to keep during updates (which includes data/covariance, pseudoinverse and projection matrices) makes it quite inefficient in terms of space and complexity.

Lu et al. presented the ICLDA [66, 67], an exact incremental version of the so-called Complete LDA [99] which is an improved version of the plain PCA+LDA approach. ICLDA works under the SSS case and obtains results equivalent to the ones obtained with the corresponding batch approach. Chunks of samples from new classes or single samples from any class are considered for the updates.

Yeh et al. [69] presented another version of the LS-ILDA from a rank-one update method with a simplified class indicator matrix, that addresses also the concept drift issue.

A new LDA variation was showed by Chun et al. in [71] as a batch (LDA/QR) and an incremental (ILDA/QR) algorithm that use the economic QR decomposition followed by solving a lower triangular system. The incremental algorithm assumes that class means remain unchanged and that all training samples are linearly independent. ILDA/QR can handle both single samples or chunks of samples both from old and new classes.

Zhang et al. [73] proposed an Incremental Regularized Least Square (IRLS) to develop an incremental LDA called LDA-IRLS. This is capable of updating the solution to the RLS problem with multiple columns on the right-hand side when a new data is acquired. LDA-IRLS works independently of any relation between the dimension and the number of training samples, and the results are equivalent to the ones obtained with the corresponding batch approach.

4.3. Discussion

Table 6 shows the characteristics of the main ILDA methods considered, in chronological order, as Table 2. In the case of chunk size-lab, the symbol • is used to denote those methods that allow updating existing classes with a new single sample, the symbol ○ is for the methods that allow updating existing classes or an unknown class with a single sample, $\overset{\bullet}{\bullet\circ}$ indicates those

Table 6: Overview of the Incremental LDA algorithms

Author	Year	Acronym	mean update	Subspace model	Chunk size-lab	size/dim ratio	Application
Pang [59, 60]	2005	CILDA	✓	EVD	○ $\overset{\bullet}{\bullet\circ}$	<	Data streams
Ye [62]	2005	IDR/QR	✓	EVD	○		Face recognition
Kim [64, 65]	2007/11	IncLDA	✓	EVD	$\overset{\bullet}{\bullet\circ}$		Object/Face recognition
Uray [68]	2007	ILDAaPCA	✓	EVD	○		Image classification
Song [70]	2008	IWAS	✓	EVD	○	>	Face recognition
Zhao [61]	2008	GSVD-ILDA	✓	SVD	$\overset{\bullet}{\bullet\circ}$		Face recognition
Zheng [72]	2009	IDSLDA	✓	EVD	● ○	<	Face recognition
Liu [63]	2009	LS-ILDA	✓	Least Squares	○		Face recognition
Lu [66, 67]	2012	ICLDA	✓	SVD	● $\overset{\circ}{\circ\circ}$	<	Face recognition
Lamba [74]	2012	ISDA	✓	EVD	$\overset{\bullet}{\bullet\circ}$		Face recognition
Peng [6]	2013	c-QR/IncLDA	✓	EVD	$\overset{\bullet}{\bullet\circ}$		Face recognition
Yeh [69]	2013	LS-ILDA-CD	✓	Least Squares	○		Pattern classification
Lu [75]	2015	c-IDR/LDA	✓	EVD	$\overset{\bullet}{\bullet\circ}$		Face recognition
Chu [71]	2015	ILDA/QR		QR+LTLS	○ $\overset{\bullet}{\bullet\circ}$	<	Pattern classification
Dhamecha [76]	2016	ISSDA	✓	EVD	$\overset{\bullet}{\bullet\circ}$		Face recognition
Zhang [73]	2016	LDA-IRLS		LSQR [105]	○		Pattern classification

methods that allow updating existing classes or an unknown class with a chunk of samples, and $\overset{\circ}{\circ\circ}$ indicates those methods that allow updating unknown class with a chunk of samples.

Regarding the complexity of the methods, Table 7 shows in chronological order the main decompositions and the computational complexity of the ILDA approaches provided by the authors, organized according to the chunk size, as Table 3. Grey and white shaded rows show covariance-based or adaptive ILDA methods, respectively. m and c are the number of samples and classes in the initial training set, respectively, n is the number of samples in the update set.

Table 7: Main decomposition and computation complexity in chronological order, showing first the methods with correction per sample and then the chunk-based ones. Grey and white shaded rows show covariance-based or adaptive ILDA methods, respectively.

Approach	Update	Decomposition(dim)	Computational complexity
IDR/QR [62]	○	**QR** (d,c)+EVD($\widetilde{S}_w^{-1}\widetilde{S}_b$)	$O(dc + c^3)$
ILDAaPCA [68]	○	aPCA + LDA	$O(dr + (r + 1)^3)$
IWAS [70]	○	EVD(c)+EVD($\widetilde{S}_w^{-1}\widetilde{S}_b$)	$O(dc + dr + c^3)$
LS-ILDA [63]	○	Projections	$O(d\ min(m, d))$
IDSLDA [72]	●	EVD($\widetilde{S}_w^{-1}\widetilde{S}_b$)+EVD($\widetilde{S}_c$)	$O(d(r + c^2))$
ICLDA [66]	●	**QR** (d,m)+**QR** (d,m)+SVD(m-c, c)+EVD($c - 1$)	$O(dm + m^2 + c^3 + (m - c)^3)$
LS-ILDA-CD [69]	○	Projections	$O(d\ min(m, d))$
LDA-IRLS [73]	○	LSQR [105]	$O(m + d + dc)$
CILDA [59]	$\overset{\bullet}{\bullet\circ}$	EVD($\widetilde{S}_w^{-1}\widetilde{S}_b$)	$O(dm + \tilde{r}^3)$
IncLDA [65]	$\overset{\bullet}{\bullet\circ}$	EVD(d,n)+QR(d,n+1)+EVD($\tilde{r_b}$)	$O(r_{ty}^3 + r_{by}^3 + d\tilde{r_t}\tilde{r_b})$
GSVD-ILDA [61]	$\overset{\bullet}{\bullet\circ}$	QR(d,n+1)+SVD(r+n+1)+SVD(c,$\tilde{r}$)	$O(drn + dn^2 + r^3 + d\tilde{r}c)$
ICLDA [66]	$\overset{\circ}{\circ\circ}$	**QR** (d,m)+**QR** (d,m)+SVD(m-c, c)+EVD($c - 1$)	$O(dm + m^2 + c^3 + (m - c)^3)$
ISDA [74]	$\overset{\bullet}{\bullet\circ}$	EVD(d,n)+QR(d,n+1)+EVD($\tilde{r_b}$)	$O(r_{ty}^3 + r_{by}^3 + d\tilde{r_t}\tilde{r_b})$
c-QR/IncLDA [6]	$\overset{\bullet}{\bullet\circ}$	**QR** (d,c)+EVD($\widetilde{S}_w^{-1}\widetilde{S}_b$)	$O(dnc^2 + dnc + nc^3)$
c-IDR/LDA [75]	$\overset{\bullet}{\bullet\circ}$	Projections+EVD($\widetilde{S}_w^{-1}\widetilde{S}_b$)	$O(c^3 + dmc)$
ILDA/QR [71]	$\overset{\bullet}{\bullet\circ}$	QR(d,n) + Projections	$O(drn + dn^2 + dnc)$
ISSDA [76]	$\overset{\bullet}{\bullet\circ}$	EVD(d,n)+QR(d,n+1)+EVD($\tilde{r_b}$)	$O(r_{by}^3 + d\tilde{r_t}\tilde{r_b})$

r_{b_y} and r_{t_y} are the subspace ranges of the between- and total- scatter matrices in the update set, respectively, and the ranges of the resulting subspaces after the update are given by $\tilde{r_b}$ and $\tilde{r_t}$. The preserved ranges in the initial and the resulting updated subspace are marked by r and $\tilde{r}$, respectively. Finally, **QR** is used to denote a QR-updating process.

We observe that, like in IPCA, the complexity of the ILDA algorithms with one sample update involve lower complexity and variability opposed to the chunk case. Among the covariance based formulations under the same dimension restriction, IDR/QR presents less complexity than ILDAaPCA and IDSLDA unless $c > r$, and IWAS since $dr > 0$. Within the adaptive ILDA, the difference in complexity between LDA-IRLS and LS-ILDA and depends on $\min(m, d)$. In the $d >> m$ case, the complexity of LS-ILDA and ICLDA is dominated by $O(dm)$, on LDA-IRLS it is $O(dc)$, where $m > c$. From the complexity viewpoint, we conclude that the best options for updating one sample are IDR/QR $(dim >< m)$ and LDA-IRLS $(d >> M)$.

From the tables, we remark that LS-ILDA [63] approach has been shown in its covariance-based and covariance free versions with one sample update. Both approaches use simple operations within matrices, vectors and scalars, which do not have a high computational cost since no matrix multiplication is performed. The complexity of these operations is at most $O(d^2)$ for the first method, and $O(dm)$ for the second one. At each update, the number of such operations is constant since LS-ILDA always picks up the method with the lowest complexity.

From empirical evaluation, authors show that LS-ILDA is more efficient than GSVD-ILDA [61] and IncLDA [65]. Regarding ICLDA [66] this approach is lower than CILDA [59] and IncLDA [65]. However, as the rank of the total scatter matrix grows, the dimensions of internal subspace get large and the storage needs become too demanding [71]. For ILDAaPCA and IWAS, the complexity is not provided, but the authors empirically show that it has a less computational cost than the batch method. In the case of IWAS, the computational cost is also the same as the Pang's CILDA cost, but using less memory requirements. ILDA/QR [71] can update the training set with both a single sample or

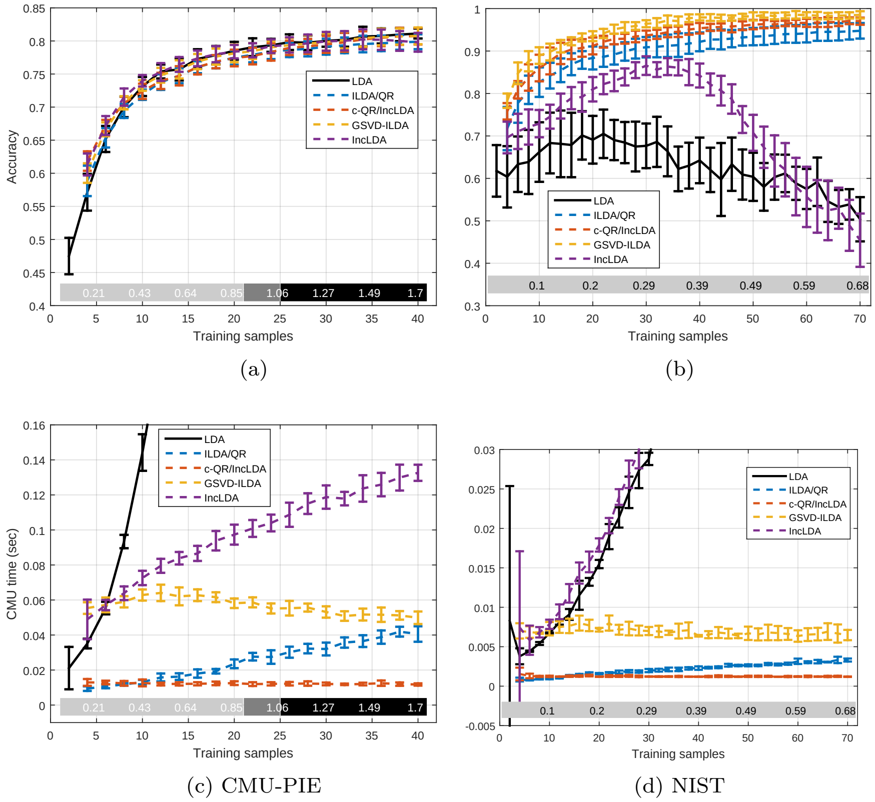
Table 8: Empirical comparison of the computation time, ILDA

→Reference ↓Under analysis	CILDA	IDR/QR	IncLDA	GSVD-ILDA	LS-ILDA	ICLDA
IncLDA [65]	< [66]	>		< [63]		
IWAS [70]	≈					
ISDA [74]	<	>	≈	<		
c-IDR/LDA [75]		<	<			
ISSDA [76]	<		<	<		
LS-ILDA [63]	< [71]	>	<	<		
LS-ILDA-CD [69]	<	>	<	<	=	
ICLDA [66]	<	> [67]	<	<		
ILDA/QR [71]	<	≈	<	<	<	<
LDA-IRLS [73]	<	≥	<	<	<	

a new chunk of samples. In the first case, the empirical results show that this is much faster than IDR/QR [62], LS-ILDA [63], and ICLDA [66]. However, when $m >> c$ the IDR/QR computational cost is smaller than ILDA/QR. The difference in running time between LS-ILDA and ILDA/QR is given by implementation or running details rather than intrinsic algorithm complexity. In the second case, ILDA/QR is faster than IncLDA [65].

Table 8 gives a comparison among those methods where empirical results concerning computational time were provided. Grey and white shaded cell show covariance-based or adaptive ILDA methods, respectively. $<$ and $>$ mean that the method in the corresponding row exhibits a smaller or higher updating time regarding the method in the corresponding column, respectively, and $\approx$ means similar or equivalent time. From the relative comparisons in the table, we can deduce that ILDA/QR and LDA-IRLS are faster than CILDA and GSVD-ILDA, which is marked by $<$, and that IDR/QR is faster than IncLDA and LS-ILDA, which is marked by $>$.

Since chunk-based ILDA algorithms involve higher complexity and variability with regard to the case of a single update, the former only has been considered to perform an empirical validation aiming at putting forward the differences and particularities some of these methods under two different cases: $d < m$ and $d > m$. The experimental setup is the same as the used in IPCA, and also the datasets used. The only difference is that CMU-PIE has been now resized to 40×40 to force the $d < m$ case. Figure 5 shows the performance and the computational cost.



(a)

(b)

(c) CMU-PIE

(d) NIST

Figure 5: Comparison of ILDA methods regarding performance and the computational cost.

We can see that the generalized methods present in general better performance than the plain LDA what is more evident when $d > m$. As to the performance and the computational cost, among the validated methods, c-QR/IncLDA [6] shows good behavior in both cases, $d > m$ and $d < m$, followed by ILDA/QR [71] and GSVD-ILDA [61].

5. Incremental Discriminative Common Vector (IDCV)

The DCV [106] approach constitutes a different way to overcome the singularity problem of LDA. It is particularly appealing because of its good performance behavior and flexibility of implementation, specially in case of very large dimensionalities such as in image recognition or genomic problems. Similarly to *null* space based methods, DCV method uses the $\mathcal{N}(S_w)$ to project the samples on it and implicitly avoid singularities. In fact, the rationale behind DCV is basically the same as with LDA and the corresponding mathematical problem can be stated as

$$\max |U^T S_b U| \quad \text{s. t. } UU^T = I \text{ and } |U^T S_w U| = 0$$

Note that this formulation only makes sense in the SSS case, and then all training data, once projected onto $\mathcal{N}(S_w)$, gets collapsed into a single vector per class that can be maximally separated from each other using only $(c-1)$ dimensions, leading to the so-called discriminant common vectors (DCVs) that can finally be used to construct any distance-based classifier. Instead of dealing with $\mathcal{N}(S_w)$, DCV uses its orthogonal complement, the *range* space, $\mathcal{R}(S_w)$ which is much smaller in the SSS case. Incremental DCV updates consist of implicitly maintaining $\mathcal{N}(S_w)$ and computing the DCVs as illustrated in Figure 6. The graphical representation of the update of the DCV models is pretty much the same as with LDA. The only but very important difference is that now the common vectors and corresponding (reduced) discriminative spaces are explicitly constructed. In the figure these are displayed as subspaces of one (two classes in the initial model) and two (three classes after the update) dimensions containing the corresponding common vectors where all the dataset (X and $\tilde{X}$, respectively) collapses.

Their good performance behavior has motivated a recent interest in obtaining efficient implementations including incremental formulations. As in previous sections, we distinguish between covariance-based and covariance-free approaches. The first type are described in subsection 5.1 and use scatter matrices and EVD. The second one are described in subsection 5.2 and use DS and orthogonalization algorithms as SVD, QR or GSO.

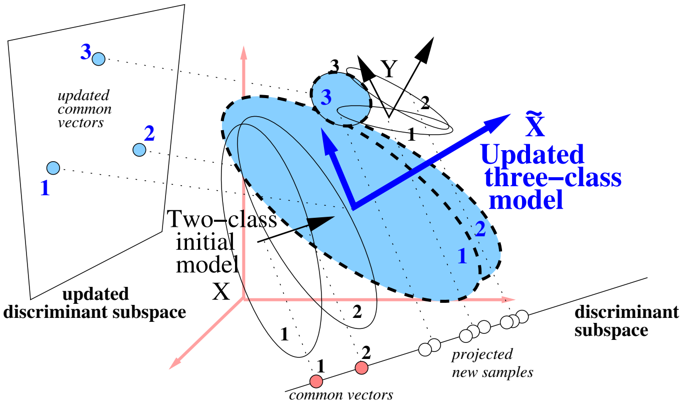


Figure 6: Subspaces and corresponding models involved in IDCV methods. The initial training set and the initial common vectors, with two classes (1 and 2), are represented using X and ○, respectively. Y is the update set with new samples for existing classes (1 and 2), as well as a new unknown class (3). $\tilde{X}$ and ○ represent the resulting updated set and the resulting updated common vectors, respectively.

5.1. Covariance-based Incremental DCV

Diaz et al. [77] stated the first covariance-based incremental DCV in which a chunk of samples is used to update all involved subspaces. The IDCV/EVDR algorithm uses a generalization of the scatter matrix decomposition in Hall et al. [44] in such a way that the $\widetilde{S}_w$ can be written as the sum of three terms: the old and new ones plus a rank-one matrix that involves their corresponding means. As a result, the updated basis that spans $\mathcal{R}(\widetilde{S}_w)$ can be obtained from the previous one, extended with new vectors obtained through projections onto $\mathcal{N}(S_w)$, and a rotation that is obtained from a new reduced eigenproblem as in Hall et al. [44]. The final projection along with DCVs is obtained through PCA in a very reduced subspace in such a way that its computational cost is almost negligible compared to the cost of updating the previous range subspace.

In [78], Diaz et al. showed that the previous formulation can be greatly improved if the method is allowed to obtain *any* equivalent basis of the *same* subspace that is obtained through the corresponding batch method. In this case, the rotation is not needed and the corresponding incremental update is faster. This algorithm is referred to as IDCV/EVD.

The above methods can only work in the SSS case as the original DCV approach. Nevertheless, Diaz et al. presented [4] an incremental version of the Rough Common Vector method of Tamura et al. [107] called Incremental Generalized Discriminative Common Vector (IGDCV) in which the conditions are relaxed by introducing the idea of approximate (extended) null and (reduced) range subspaces. IGDCV allows updating the training set with a single sample, or with a chunk of samples from existing or new classes. As a consequence of its formulation, IGDCV produces similar results as the ones in the corresponding batch version whose accuracy is controlled by keeping track of the greatest old and new eigenvalues.

Under the same previous framework, a decremental version of GDCV that removes unnecessary data and/or classes to update a previously learnt model without recalculating the full projection is presented in [81]. The authors show a considerable computational gain without compromising the accuracy of the model.

5.2. Covariance-free Incremental DCV

A first approach in which basis vectors were updated directly without managing any scatter matrix was introduced by Ferri et al. [79]. The corresponding online DCV (oDCV) algorithm, considered one single sample at a time and used only basic vector operations (differencing, projection and normalization) and an economic QR decomposition.

Diaz et al. [77] offered (along with the IDCV/EVDR algorithm) a more general alternative to the oDCV idea using an incremental GSO procedure and considering chunks of samples. In this IDCV/EVD algorithm, the DS corresponding to new data is obtained and the current basis of the range space is updated using GSO.

Lu et al. [82] presented an IDCV/QR approach that allows updates with a new single sample. The method uses rank-one QR updating and is equivalent to the previous oDCV algorithm [79]. Moreover, IDCV/QR considers the case of samples belonging to new unseen classes by introducing a GSO procedure. In both cases, the result is the same as in the original batch approach at a very

Table 9: Overview of the Incremental DCV algorithms

Author	Year	Acronym	mean update	Subspace model	Chunk size-lab	size/dim ratio	Application
Ferri [79]	2010	oDCV		QR	●	<	Face recognition
Diaz [77]	2010	IDCV/EVDR	✓	EVD	● ●○	<	Image Classification
Diaz [77]	2010	IDCV/GSO		GSO	● ● ●○	<	Image Classification
Diaz [78]	2011	IDCV/EVD	✓	EVD	● ●○	<	Image Classification
Lu [82]	2012	IDCV/QR		QR/GSO	○	<	Face recognition
Ferri [80]	2013	ITDCV		SVD	●		Image Classification
Diaz [4]	2015	IGDCV	✓	EVD	● ●○		Image Classification
Diaz [81]	2017	DGDCV	✓	EVD	●○ ●		Image Classification
Zhu [83]	2017	IOCA		GSO	●	<	Pattern classification

reduced computational cost.

Ferri et al. [80] approximated the DCV by using rank one SVD updates in an incremental formulation. The algorithm consists of a correction per sample along with an additional restriction on the growth of the range space, $\mathcal{R}(\widetilde{S}_w)$. The main difference among the above approaches is that the importance of each sample at an update is measured by its norm after projection. Using a dynamic thresholding, the algorithm both applies the correction to the subspaces and decides about whether or not to increase/decrease the corresponding range/null space.

Equivalent to the previous oDCV [79] and IDCV/QR [82], Zhu et al. [83] exploit the implicit incrementality of GSO to introduce an Incremental Orthogonal Component Analysis (IOCA) by using DS. IOCA automatically extracts desired orthogonal components using an adaptive threshold policy.

5.3. Discussion

The main features of the IDCV methods are summarized in Table 9. Its structure is the same as in Table 6. It can be noticed how covariance-based methods have an exact mean update while covariance free methods do not, since the later are based on the projections into the DS followed by the orthogonalization process, in such a way that the subtrahend vector does not change and the updated mean is not necessary.

Table 10 shows in chronological order the main decomposition and computational complexity of the IDCV approaches provided by the authors, like Table 7. Grey and white shaded rows show covariance-based or covariance-free

IDCV methods, respectively. r_w, r_{w_y} and $\tilde{r_w}$ are the ranges of the within-scatter matrices in the initial, update, and resulting subspaces, respectively. The preserved ranges in the initial DS and the resulting DS are denoted by r and $\tilde{r}$, respectively.

As in the previous incremental approaches, the complexity of the IDCV algorithms with one sample update involve low complexity and variability than when considering chunk updates, with a complexity dominated by $O(dr)$. The difference in computational cost between oDCV [79] and IDCV/QR [82] is a reduction in the quadratic term for the number of classes. The authors of oDCV and ITDCV, show that these present better discriminant characteristics, and less complexity than LS-ILDA [63], and that IDCV/QR has better performance than IDR/QR [62], with a similar computational cost.

Regarding the IDCV/EVDR and the IDCV/GSO algorithms, by Diaz et al. [77], IDCV/GSO gets significantly higher savings than IDCV/EVDR in the first iteration. However, this is only true in case of small datasets, i.e. small m values. On the contrary, IDCV/EVDR is more efficient than IDCV/GSO (with regard to its batch counterpart) for larger values of m. This behavior gets more evident in the case of larger databases.

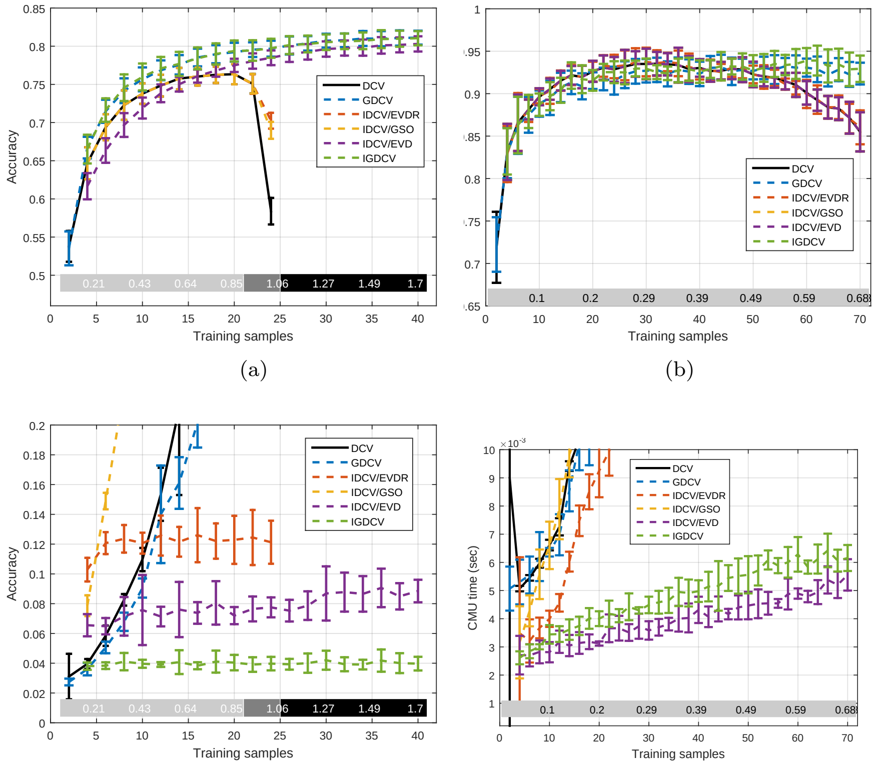
The computational complexity of IDCV/EVD [78] decreases with m for a fixed n value. The authors assert that this tendency follows approximately the theoretical $O(1/m)$. So when $m \gg n$, the differences in computational cost are more significant with respect to the batch implementation. If $m \approx n$, the computational cost of incremental algorithm surpasses the batch one.

Table 10: Main decomposition and computation complexity. Grey and white shaded rows show covariance-based or covariance-free IDCV methods, respectively.

Approach	Update	Decomposition(dim)	Computational complexity
oDCV [79]	●	Norm(d,1)+QR(d,c)	$O(dr + dc^2)$
IDCV/GSO [77]	●	Norm(d,1)+GSO(d,c)	$O(dr)$
IDCV/QR [82]	○	Norm(d,1)+QR(d,c)	$O(dr + dc)$
ITDCV [80]	●	SVD(d,r+1)	$O(dr + r^3)$
IOCA [83]	●	Norm(d,1)+GSO(d,r)	$O(dr)$
IDCV/EVDR [77]	● ●○	EVD(n)+GSO(d,r_{w_y}+c)+EVD($\tilde{r_w}$)	$O(n^3 + \tilde{r_w}^3 + d(n^2 + \tilde{r_w}^2))$
IDCV/GSO [77]	● ●○	GSO(d,n)+GSO(d,c)	$O(dn^2)$
IDCV/EVD [78]	● ●○	EVD(n)+GSO(d,r_{w_y}+c)	$O(n^3 + dmn)$
IGDCV [4]	● ●○	GSO(d,n+c)+EVD($r_w + r_{w_y}$)	$O(dr_w(n + c) + nr_w^2 + (r_w + r_{w_y})^3 + d\tilde{r_w}^2)$

For IGDCV [4], authors show that the larger the difference in size between the already processed set and the data being incorporated is, the greater the 720 difference in computational cost with respect to the batch algorithm.

Like in the previous subsection, an empirical validation has been carried out to show the performance and complexity in practice of some of the considered methods under two cases: $d < m$ and $d > m$, when applying chunk updates. Figure 7 shows the performance and the computational cost, with $\alpha = 0.05$ 725 to GDCV method. We can see that the generalized method, exhibits better performance than the plain DCV method, and that this difference it is more evident when $d < m$. In terms of performance and computational cost, the IGDCV method [4] shows the best behavior among the validated methods.



(a)

(b)

(c) CMU-PIE

(d) NIST

Figure 7: Comparison of IDCV methods regarding performance and the computational cost.

6. Performance analysis

This section gives an overview of the discriminant properties, in terms of accuracy rate, according to the empirical results provided by the authors of the incremental algorithms referenced in this paper. The evaluation and comparison of the performance of the incremental approaches is a difficult task, since few of them contain numerical results. As well, these results are usually not comparable across publications largely due to different experimental setups. For instance, ICLDA [66] and LDA/QR [71] with a single sample update presents an accuracy rate of 84.9 ± 1.2 and $70.28\% \pm 1.46$, respectively, on FERET. Although the same dataset is used and in both approaches the update is performed with a single new sample, those numbers are not directly comparable. In ICLDA the initial training set contains 180 of 200 classes, such that the update set in each step may update an existing class or create a new class. The 1-NN using cosine distance is used as classifier. In ILDA/QR, the initial training set has 200 of 200 classes, such that the single new sample always belongs to one of the existing classes. In this case the 1-NN applying euclidean distance is used as classifier.

Table 11 presents an overview of the comparative performance in terms of their classification accuracy. We report the comparisons among the methods, their batch versions and other methods cited in the corresponding paper. Since most of the results depicted by the authors are graphical rather than numerical, the articles that present numerical results are indicated with shaded background. First column shows the approach under analysis, second column the comparison against its batch version, and $\gg$, $>$, $\geq$, $\approx$, $=$ and $<$ mean that it exhibits much better, better, better or same, similar, same and less accuracy than the method in the column.

Regarding the incremental PCA, Table 11 shows that the worst approaches compared to batch methods are CCIPCA [42] and LET-IPCA [53]. It can also be inferred that the MSES approach of Hall et. al in [44] and the EVVD-IPCA [58]

Table 11: Empirical comparison in terms of accuracy for the incremental approaches

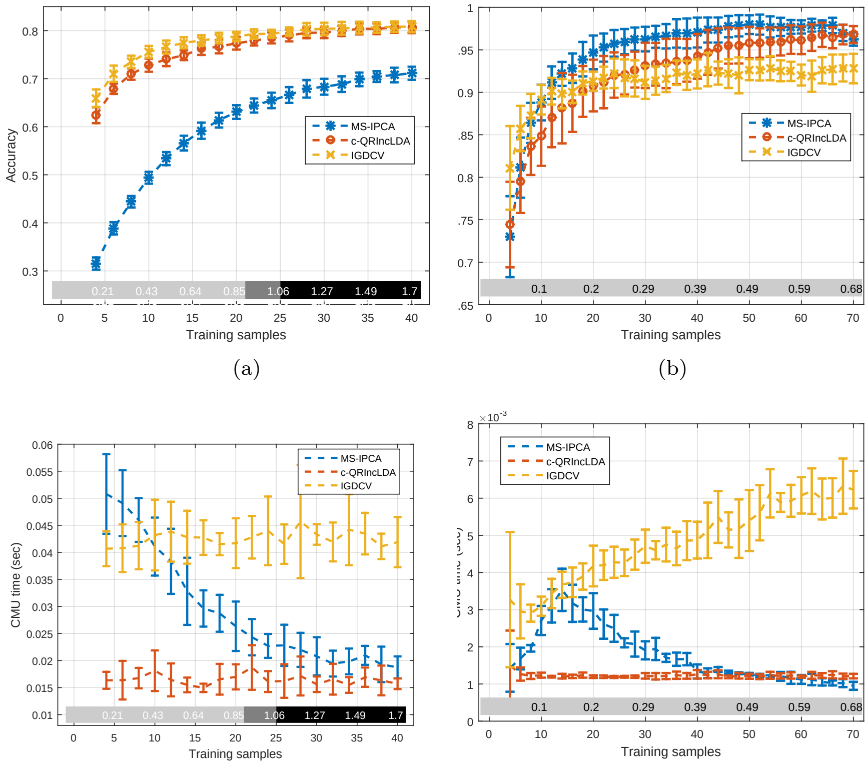
Approach	Batch	≫	>	≥	≈	=
IPCA						
CCIPCA [42]	< [53]					
MMSES [47]	≈					
LET-IPCA [53]	<		CCIPCA			
SVDU-IPCA [52]	≈		CCIPCA			
MMSES-C [11]	≈					
MS-IPCA [54]	≈		CCIPCA		SVDU-IPCA, MSES	
RIPCA-C [56]	≈		RIPCA			
AIPCA [55]	=			CCIPCA		
SA-IPCA [57]	≈				CCIPCA [108]	
EVVD-IPCA [58]	=					MSES
ILDA						
CILDA [59]	≈	MSES	IncLDA [66]			
IDR/QR [62]	<					
IncLDA [65]	≈					
ILDAaPCA [68]	≈					
IWAS [70]		RIPCA			CILDA	
GSVD-ILDA [61]	=	CILDA	IDR/QR			
IDSLDA [72]	≈		IDR/QR			
LS-ILDA [63]	≈ [69]			IncLDA	LS-ILDA-CD [69]	GSVD-ILDA
ICLDA [66]	=	IDR/QR [67], IncLDA	CILDA			
ISDA [74]	<	CCIPCA	IncLDA			
c-QR/IncLDA [6]				IDR/QR		
LS-ILDA-CD [69]	≈				LS-ILDA	
c-IDR/QR [75]	=	IDR/QR IncLDA				
ILDA/QR [71]	=					
ISSDA [76]	≈	MSES		IncLDA		
LDA-IRLS [73]	=	IncLDA, LS-ILDA	IDR/QR			
IDCV						
oDCV [79]	=			LS-ILDA		
IDCV/EVDR [77]	≈					
IDCV/GSO [77]	≈					
IDCV-EVD [78]	=					
IDCV/QR [82]	=			IDR/QR		
ITDCV [80]	≈		LS-ILDA		oDCV	
IGDCV [4]	≈					
IOCA [83]	=					< CCIPCA

have better performance than CCIPCA, and that EVVD-IPCA could present the same or better discriminant properties than MS-IPCA [54] and SVDU-IPCA [52].

In the incremental LDA approaches, IDR/QR [62] presents less discriminant properties with regard to its batch approach. Conversely, GSVD-ILDA [61], ICLDA [66] and ILDA/QR [71] have an exact approximation to the batch method. From the table we deduce that IWAS [70] and GSVD-ILDA have better performance than the IncLDA [65], and that LS-ILDA [63] improve the results by CILDA [59] and IDR/QR.

Regarding the incremental DCV approaches, oDCV [79], IDCV/EVD [78], and IDCV/QR [82], have an exact approximation to its batch method, and we can deduce that these methods have better performance than IncLDA, CILDA and IDR/QR.

From the empirical validation made in this document the MS-IPCA [54], c-QR/IncLDA [6] and IGDCV [4] methods have been highlighted, from performance and computational cost viewpoint. Figure 8 shows the performance and the computational cost of these methods, under the same experimental setup 775 used in ILDA and IDCV. In both cases $d > m$ and $d < m$, c-QR/IncLDA [6] shows the best performance regarding computational cost.



(a)

(b)

(c) CMU-PIE

(d) NIST

Figure 8: Comparison of the MS-IPCA, c-QR/IncLDA and IGDCV methods regarding performance and the computational cost.

7. Summary and concluding remarks

An exhaustive survey on incremental feature extraction based on linear sub- 780 space methods with orthogonal matrix constraints based on global loss function has been presented in this paper. Incremental methods are discussed and categorised in terms of subspace model, decomposition of matrices, updating strategy, requirements to be applied, computational complexity, experimental setup and accuracy rates. These different aspects allow us to analyze their properties and suitability when applied to classification.

In particular, our survey focuses on the incremental approaches of the PCA, LDA and DCV batch methods, which are able to incorporate new information to the acquired knowledge, without training the system from scratch. We categorized these approaches in *covariance-based* and *covariance-free* methods, as well as in the possible updating strategies, from updating with one single sample or chunks of new data at each incremental step, to updating with samples associated to previous class labels or to new ones. Differentiation between approaches working under or addressing the *SSS* problem is also established.

Regarding IPCA algorithms, three types of categorization were held due to the large variety of approaches. First, *covariance-based* approaches, whose aim is to maintain and update a more or less explicit model of the scatter matrix using mainly EVD. Second, *SVD updates* based on partial SVD updates that modify principal components without constructing or referring to a covariance-like matrix. Third and last, *covariance-free* based approaches. For ILDA, two main categories were held, *covariance-based* methods, which comprise those algorithms that follow the classic LDA rationale and are based on an EVD of an appropriate matrix; and *adaptive* ILDA, which use either local learning rules, adapted SVD or QR decompositions to update the subspace models. Finally, IDCV algorithms are also categorised in two types, *covariance-based* and that *covariance-free*. The first type uses scatter matrices and EVD. The second type uses DS and some orthogonalization algorithm such as SVD, QR or GSO.

Given the importance of space and computational complexity as indicators to evaluate the efficiency of the algorithms, we performed a comparative study between the different algorithms according to the data provided by different authors. This comparison is established in terms of computational complexity, the main decomposition employed, the experimental setup and its accuracy rate.

As well, we have carried on several empirical experiments to compare among some of the presented algorithms. In this way, we have found that MS-IPCA [54], c-QR/IncLDA [6] and IGDCV [4] highlight from performance and computational cost viewpoint, which corroborates the conclusions derived from the literature analysis in Table 11 and the computation time comparison tables.

As future works of the incremental feature extraction is aimed at generating efficient algorithms in real time without any restrictions about the dimension and the number of training samples and classes. In addition, the incremental concept is being extended to the decremental and dual form. Finally, the explosion of deep learning opens new opportunities to take leverage of these incremental schemas to be applied onto pre-trained neural networks, where embeddings and subspaces created with autoencoders can be enhanced and extended incrementally using transfer learning. In particular cross-class transfer learning poses an opportunity to combine incremental subspace-learning techniques with deep learning that may reduce the vast amounts of required data.

References

[1] G. Chen, W. Tsai, An incremental-learning-by-navigation approach to vision-based autonomous land vehicle guidance in indoor environments using vertical line information and multiweighted generalized hough transform technique, IEEE Trans. Systems, Man, and Cybernetics (Part B) 28 (5) (1998) 740–748. doi:10.1006/cviu.1999.0746.

[2] M. Pardowitz, S. Knoop, R. Dillmann, R. Zöllner, Incremental learning of tasks from user demonstrations, past experiences, and vocal comments., IEEE Trans. Systems, Man, and Cybernetics (Part B) 37 (2) (2007) 322–332. doi:10.1109/TSMCB.2006.886951.

[3] D. Ross, J. Lim, R. Lin, M. Yang, Incremental learning for robust visual tracking, IJCV 77 (1-3) (2008) 125–141. doi:10.1007/s11263-007-0075-7.

[4] K. Diaz-Chito, F. Ferri, W. Díaz-Villanueva, Incremental generalized discriminative common vectors for image classification, IEEE Trans. Neural Networks and Learning Systems 26 (8) (2015) 1761–1775. doi:10.1109/TNNLS.2014.2356856.

[5] X. Zeng, G. Li, Covariance free incremental principal component analysis with exact mean update, Journal of Computational Information Systems 5 (16) (2013) 181–192.

[6] Y. Peng, S. Pang, G. Chen, A. Sarrafzadeh, T. Ban, D. Inoue, Chunk incremental IDR/QR LDA learning, in: IJCNN, 2013, pp. 1–8. doi:10.1109/IJCNN.2013.6707018.

[7] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, Journal of machine learning research 10 (Mar) (2009) 623–656.
URL http://dl.acm.org/citation.cfm?id=1577069.1577091

[8] E. Diaz-Aviles, L. Drumonds, L. Schmidt-Thieme, W. Nejdl, Real-time top-n recommendation in social streams, in: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, 2012, pp. 59–66. doi:10.1145/2365952.2365968.

[9] J. Vinagre, J. Alípio, J. J. Gama, Fast incremental matrix factorization for recommendation with positive-only feedback, in: User Modeling, Adaptation, and Personalization: 22nd International Conference, UMAP 2014, 2014, pp. 459–470. doi:10.1007/978-3-319-08786-3_41.

[10] Q. Song, J. Cheng, H. Lu, Incremental matrix factorization via feature space re-learning for recommender system, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, 2015, pp. 277–280. doi:10.1145/2792838.2799668.

[11] S. Ozawa, S. Pang, N. Kasabov, On-line feature selection for adaptive evolving connectionist systems, IJICIC (2006) 181–192.

[12] A. Levy, M. Lindenbaum, Sequential karhunen-loeve basis extraction and its application to images, IEEE Transactions on Image Processing 9 (8) (2000) 1371–1374. doi:10.1109/83.855432.

[13] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowledge-Based Systems 27 (2012) 271–280. doi:https://doi.org/10.1016/j.knosys.2011.09.006.

[14] G. Ling, H. Yang, I. King, M. R. Lyu, Online learning for collaborative filtering, in: The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, pp. 1–8. doi:10.1109/IJCNN.2012.6252670.

[15] C. Zhang, H. Wang, S. Yang, Y. Gao, Incremental nonnegative matrix factorization based on matrix sketching and k-means clustering, in: Intelligent Data Engineering and Automated Learning – IDEAL 2016: 17th International Conference, 2016, pp. 426–435. doi:10.1007/978-3-319-46257-8_46.

[16] X. Huang, L. Wu, E. Chen, H. Zhu, Q. Liu, Y. Wang, Incremental matrix factorization: A linear feature transformation perspective, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 1901–1908. doi:10.24963/ijcai.2017/264.

[17] T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, Pcanet: A simple deep learning baseline for image classification?, IEEE Transactions on Image Processing 24 (12) (2015) 5017–5032. doi:10.1109/TIP.2015.2475625.

[18] H. Zheng, W. Cai, T. Zhou, S. Zhang, M. Li, Text-independent voice conversion using deep neural network based phonetic level features, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 2872–2877. doi:10.1109/ICPR.2016.7900072.

[19] M. Seuret, M. Alberti, R. Ingold, M. Liwicki, Pca-initialized deep neural networks applied to document image analysis, CoRR abs/1702.00177.
URL http://arxiv.org/abs/1702.00177

[20] F. Xu, G. Gu, X. Kong, P. Wang, K. Ren, Object tracking based on two-dimensional pca, Optical Review 23 (2) (2016) 231–243. doi:10.1007/s10043-015-0178-2.

[21] C. Bo, D. Wang, Online object tracking based on convex hull representation, in: Parallel and Distributed Systems (ICPADS), 2016 IEEE 22nd International Conference on, 2016, pp. 1221–1224. doi:10.1109/ICPADS.2016.0164.

[22] P. Liang, Y. Wu, H. Ling, Planar object tracking in the wild: A benchmark, arXiv preprint arXiv:1703.07938.

[23] L. Sun, C. Hsiung, C. G. Pederson, P. Zou, V. Smith, M. von Gunten, N. A. O'Brien, Pharmaceutical raw material identification using miniature near-infrared (micronir) spectroscopy and supervised pattern recognition using support vector machine, Applied spectroscopy 70 (5) (2016) 816–825. doi:10.1177/0003702816638281.

[24] D. R. Vidhyavathi, Principal component analysis (pca) in medical image processing using digital imaging and communications in medicine (dicom) medical images, International Journal of Pharma and Bio Sciences 8 (2017) 598–606. doi:10.22376/ijpbs.2017.8.2.b.598-606.

[25] D. Nandi, A. S. Ashour, S. Samanta, S. Chakraborty, M. A. Salem, N. Dey, Principal component analysis in medical image processing: a study, International Journal of Image Mining 1 (1) (2015) 65–86. doi:10.1504/IJIM.2015.070024.

[26] P. Boissard, M. Vincent, M. Sabine, A cognitive vision approach to early pest detection in greenhouse crops, computers and electronics in agriculture 62 (2) (2008) 81–93. doi:https://doi.org/10.1016/j.compag.2007.11.009.

[27] Y. Wang, M. Weyrich, An adaptive image processing system based on incremental learning for industrial applications, in: Emerging Technology and Factory Automation (ETFA), 2014 IEEE, 2014, pp. 1–4. doi:10.1109/ETFA.2014.7005346.

[28] E. Robotti, E. Marengo, Chemometric multivariate tools for candidate biomarker identification: Lda, pls-da, simca, ranking-pca, 2-D PAGE Map Analysis: Methods and Protocols (2016) 237–267doi:10.1007/978-1-4939-3255-9_14.

[29] K. Diaz-Chito, G. Konstantia, K. Anastasios, J. M. del Rincon, Incremental model learning for spectroscopy-based food analysis, Chemometrics and Intelligent Laboratory Systems 167 (2017) 123–131. doi:https://doi.org/10.1016/j.chemolab.2017.06.002.

[30] H. Hamid, F. Zainon, T. P. Yong, Performance analysis: An integration of principal component analysis and linear discriminant analysis for a very large number of measured variables, Research Journal of Applied Sciences (2016) 1422–1426doi:10.3923/rjasci.2016.1422.1426.

[31] A. D. Bimbo, D. Fabrizio, L. Giuseppe, A real time solution for face logging (2009) 1–6doi:10.1049/ic.2009.0238.

[32] A. Usman, S. Ahmed, J. Ferzund, A. Mehmood, A. Rehman, Using pca and factor analysis for dimensionality reduction of bio-informatics data, arXiv preprint arXiv:1707.07189doi:10.14569/IJACSA.2017.080551.

[33] W. Ni, S. Tan, W. Ng, S. Brown, Localized, adaptive recursive partial least squares regression for dynamic system modeling, Industrial & Engineering Chemistry Research 51 (23) (2012) 8025–8039. doi:10.1021/ie203043q.

[34] H. Nakayama, T. Harada, Y. Kuniyoshi, Ai goggles: Real-time description and retrieval in the real world with online learning, in: Computer and Robot Vision, 2009. CRV'09. Canadian Conference on, 2009, pp. 184–191. doi:10.1109/CRV.2009.9.

[35] J. Sun, D. Tao, S. Papadimitriou, P. Yu, C. Faloutsos, Incremental tensor analysis: Theory and applications, ACM Trans. Knowl. Discov. Data 2 (3) (2008) 11:1–11:37. doi:10.1145/1409620.1409621.

[36] A. A. Sobral, C. Baker, T. Bouwmans, E. Zahzah, Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction, in: Image Analysis and Recognition: 11th International Conference, ICIAR 2014, 2014, pp. 94–103. doi:10.1007/978-3-319-11758-4_11.

[37] W. Zhang, H. Sun, X. Liu, Xiaohui, Guo, An incremental tensor factorization approach for web service recommendation, in: 2014 IEEE International Conference on Data Mining Workshop, 2014, pp. 346–351. doi:10.1109/ICDMW.2014.176.

[38] E. Gujral, R. Pasricha, E. Papalexakis, Sambaten: Sampling-based batch incremental tensor decomposition, CoRR abs/1709.00668.
URL http://arxiv.org/abs/1709.00668

[39] L. Mo, Z. Feng, J. Qian, Human daily activity recognition with wearable sensors based on incremental learning, in: Sensing Technology (ICST), 2016 10th International Conference on, 2016, pp. 1–5. doi:10.1109/ICSensT.2016.7796224.

[40] H. Murakami, B. Kumar, Efficient calculation of primary images from a set of images, IEEE Trans. Pattern Anal. Mach. Intell. 4 (5) (1982) 511–515. doi:10.1109/TPAMI.1982.4767295.

[41] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler, H. Zhang, An eigenspace update algorithm for image analysis, Graphical Models and Image Processing 59 (5) (1997) 321–332. doi:10.1006/gmip.1997.0425.

[42] J. Weng, Y. Zhang, W. Hwang, Candid covariance-free incremental principal component analysis, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003) 1034–1040. doi:10.1109/TPAMI.2003.1217609.

[43] P. Hall, D. Marshall, R. Martin, Incremental eigenanalysis for classification, in: British Machine Vision Conference, 1998, pp. 286–295. doi:10.5244/C.12.29.

[44] P. Hall, D. Marshall, R. Martin, Merging and splitting eigenspace models, IEEE Trans. Pattern Anal. Mach. Intell. 22 (9) (2000) 1042–1049. doi:10.1109/34.877525.

[45] P. Hall, D. Marshall, R. Martin, Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition, Image and Vision Computing 20 (13-14) (2002) 1009–1016. doi:10.1016/S0262-8856(02)00114-2.

[46] D. Skoĉaj, A. Leonardis, Weighted and robust incremental method for subspace learning, in: 9th IEEE International Conference on CV, Vol. 2, 2003, pp. 1494–1501. doi:10.1109/ICCV.2003.1238667.

[47] S. Ozawa, S. Pang, N. Kasabov, A modified incremental principal component analysis for on-line learning of feature space and classifier, in: PRICAI: Trends in Artificial Intelligence, Vol. 3157, 2004, pp. 231–240. doi:10.1007/978-3-540-28633-2_26.

[48] S. Ozawa, S. Pang, N. Kasabov, Incremental learning of chunk data for online pattern classification systems, IEEE Trans. Neural Networks 19 (6) (2008) 1061–1074. doi:10.1109/TNN.2007.2000059.

[49] J. Kwok, H. Zhao, Incremental eigen decomposition, in: International Conference on Artificial Neural Networks, 2003, pp. 270–273.

[50] X. Qu, M. Yao, Adaptive subspace incremental PCA based online learning for object classification and recognition, in: 2011 4th International Congress on Image and Signal Processing, Vol. 3, 2011, pp. 1494–1498. doi:10.1109/CISP.2011.6100435.

[51] Y. Li, On incremental and robust subspace learning, Pattern Recognition 37 (2004) 1509–1518. doi:10.1016/j.patcog.2003.11.010.

[52] H. Zhao, P. Yuen, J. T. Kwok, A novel incremental principal component analysis and its application for face recognition, IEEE Trans. Systems, Man, and Cybernetics (Part B) 36 (2006) 873–886. doi:10.1109/TSMCB.2006.870645.

[53] S. Yan, X. Tang, Largest-eigenvalue-theory for incremental principal component analysis, in: IEEE ICIP, Vol. 1, 2005, pp. I–1181–4. doi:10.1109/ICIP.2005.1529967.

[54] D. Huang., Y. Zhang, P. Xiaorong, A new incremental PCA algorithm with application to visual learning and recognition, Neural Processing Letters 30 (3) (2009) 171–185. doi:10.1007/s11063-009-9117-1.

[55] W. Li, C. Shuo, W. Chengdong, An accurate incremental principal component analysis method with capacity of update and downdate, in: International Conference on Computer Science and Information Technology, Vol. 51, 2012, pp. 118–123. doi:10.7763/IPCSIT.2012.V51.21.

[56] G. Duan, Y. Chen, Batch-incremental principal component analysis with exact mean update, in: 18th IEEE ICIP, 2011, pp. 1397–1400. doi:10.1109/ICIP.2011.6115700.

[57] R. Arora, A. Cotter, K. Livescu, N. Srebro, Stochastic optimization for PCA and PLS, in: Allerton Conference on Communication, Control, and Computing (Allerton), 2012, pp. 861–868. doi:10.1109/Allerton.2012.6483308.

[58] B. Jin, Z. Jing, H. Zhao, EVD dualdating based online subspace learning, Mathematical Problems in Engineering 429451 (2014) 21. doi:10.1155/2014/429451.

[59] S. Pang, S. Ozawa, N. Kasabov, Incremental linear discriminant analysis for classification of data streams, IEEE Trans. Systems, Man, and Cybernetics (Part B) 35 (5) (2005) 905–914. doi:10.1109/TSMCB.2005.847744.

[60] S. Pang, S. Ozawa, N. Kasabov, Chunk incremental LDA computing on data streams, in: Advances in Neural Networks ISNN 2005, Vol. 3497, 2005, pp. 51–56. doi:10.1007/11427445_9.

[61] H. Zhao, P. Yuen, Incremental linear discriminant analysis for face recognition, IEEE Trans. Systems, Man, and Cybernetics (Part B) 38 (1) (2008) 210–221. doi:10.1109/TSMCB.2007.908870.

[62] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, V. Kumar, IDR/QR: An incremental dimension reduction algorithm via qr decomposition, IEEE Trans. Knowledge and Data Engineering 17 (9) (2005) 1208–1222. doi:10.1109/TKDE.2005.148.

[63] L. Liu, Y. Jiang, Z. Zhou, Least square incremental linear discriminant analysis, in: 9th IEEE ICDM, 2009, pp. 298–306. doi:10.1109/ICDM.2009.78.

[64] T. Kim, K. Kenneth, B. Stenger, J. Kittler, R. Cipolla, Incremental linear discriminant analysis using sufficient spanning set approximations, in: IEEE Conference on CVPR '07., 2007, pp. 1–8. doi:10.1109/CVPR.2007.382985.

[65] T. Kim, B. Stenger, J. Kittler, R. Cipolla, Incremental linear discriminant analysis using sufficient spanning sets and its applications, IJCV 91 (2) (2011) 216–232. doi:10.1007/s11263-010-0381-3.

[66] G. Lu, J. Zou, Y. Wang, Incremental complete LDA for face recognition, Pattern Recognition 45 (7) (2012) 2510–2521. doi:10.1016/j.patcog.2012.01.018.

[67] G. Lu, J. Zou, Y. Wang, Incremental learning of complete linear discriminant analysis for face recognition, Knowl.-Based Syst. 31 (2012) 19–27. doi:10.1016/j.knosys.2012.01.016.

[68] M. Uray, D. Skocaj, P. M. Roth, H. Bischof, A. Leonardis, Incremental LDA learning by combining reconstructive and discriminative approaches, in: British Machine Vision Conference, 2007, pp. 44.1–44.10.

[69] Y.-R. Yeh, Y.-C. F. Wang, A rank-one update method for least squares linear discriminant analysis with concept drift, Pattern Recognition 46 (5) (2013) 1267 – 1276. doi:10.1016/j.patcog.2012.11.008.

[70] F. Song, H. Liu, D. Zhang, J. Yang, A highly scalable incremental facial feature extraction method, Neurocomputing 71 (10–12) (2008) 1883–1888. doi:10.1016/j.neucom.2007.09.022.

[71] D. Chu, L. Liao, M. Ng, X. Wang, Incremental linear discriminant analysis: A fast algorithm and comparisons, IEEE Trans. Neural Networks and Learning Systems 26 (11) (2015) 2716–2735. doi:10.1109/TNNLS.2015.2391201.

[72] W. Zheng, X. Tang, Fast algorithm for updating the discriminant vectors of dual-space LDA, IEEE Trans. Information Forensics and Security 4 (3) (2009) 418–427. doi:10.1109/TIFS.2009.2025844.

[73] X. Zhang, L. Cheng, D. Chu, L. Liao, M. K. Ng, R. C. E. Tan, Incremental regularized least squares for dimensionality reduction of large-scale data, SIAM Journal on Scientific Computing 38 (3) (2016) B414–B439. doi:10.1137/15M1035653.

[74] H. Lamba, T. I. Dhamecha, M. Vatsa, R. Singh, Incremental subclass discriminant analysis: A case study in face recognition, in: 19th IEEE International Conference on Image Processing, 2012, pp. 593–596. doi:10.1109/ICIP.2012.6466929.

[75] G.-F. Lu, Z. Jian, Y. Wang, Incremental learning from chunk data for IDR/QR, Image and Vision Computing 36 (2015) 1 – 8. doi:10.1016/j.imavis.2015.01.002.

[76] T. I. Dhamecha, R. Singh, M. Vatsa, On incremental semi-supervised discriminant analysis, Pattern Recognition 52 (2016) 135 – 147. doi:10.1016/j.patcog.2015.09.030.

[77] K. Diaz-Chito, F. Ferri, W. Díaz-Villanueva, Image recognition through incremental discriminative common vectors, in: Advanced Concepts for Intelligent Vision Systems - ACIVS, 2010, pp. 304–311. doi:10.1007/978-3-642-17691-3_28.

[78] K. Diaz-Chito, F. Ferri, W. Díaz-Villanueva, Null space based image recognition using incremental eigendecomposition, in: Pattern Recognition and Image Analysis, Vol. 6669, 2011, pp. 313–320. doi:10.1007/978-3-642-21257-4_39.

[79] F. Ferri, K. Diaz-Chito, W. Díaz-Villanueva, Efficient dimensionality reduction on undersampled problems through incremental discriminative common vectors, in: 10th IEEE ICDM Workshops, 2010, pp. 1159–1166. doi:10.1109/ICDMW.2010.50.

[80] F. Ferri, K. Diaz-Chito, W. Díaz-Villanueva, Fast approximated discriminative common vectors using rank-one SVD updates, in: Neural Information Processing, Vol. 8228, 2013, pp. 368–375. doi:10.1007/978-3-642-42051-1_46.

[81] K. Diaz-Chito, J. M. del Rincn, A. Hernndez-Sabat, Decremental generalized discriminative common vectors applied to images classification, Knowledge-Based Systems 131 (2017) 46 – 57. doi:http://dx.doi.org/10.1016/j.knosys.2017.05.020.

[82] G. Lu, J. Zou, Y. Wang, Incremental learning of discriminant common vectors for feature extraction, Applied Mathematics and Computation 218 (22) (2012) 11269–11278. doi:10.1016/j.amc.2012.05.019.

[83] T. Zhu, Y. Xu, F. Shen, J. Zhao, An online incremental orthogonal component analysis method for dimensionality reduction, Neural Networks 85 (2017) 33 – 50. doi:10.1016/j.neunet.2016.10.001.

[84] G. Golub, Some modified matrix eigenvalue problems, SIAM Review 15 (2) (1973) 318–334. doi:10.1137/1015032.

[85] P. Gill, G. Golub, W. Murray, M. Saunders, Methods for modifying matrix factorizations, Mathematics of Computation 28 (1974) 505–535.

[86] J. Bunch, P. Nielsen, D. Sorensen, Rank-one modification of the symmetric eigenproblem, Numerische Mathematik 31 (1) (1978) 31–48. `doi:10.1007/BF01396012`.

[87] J. Bunch, P. Nielsen, Updating the singular value decomposition, Numerische Mathematik 31 (2) (1978) 111–129. `doi:10.1007/BF01397471`.

[88] M. Brand, Incremental singular value decomposition of uncertain data with missing values, in: European Conference on Computer Vision-Part I, 2002, pp. 707–720. `doi:10.1007/3-540-47969-4_47`.

[89] G. H. Golub, C. F. V. Loan, Matrix computations (3. ed.), Johns Hopkins University Press, 1996.

[90] H. Zha, H. D. Simon, On updating problems in latent semantic indexing, SIAM J. Scientific Computing 21 (2) (1999) 782–791. `doi:10.1137/S1064827597329266`.

[91] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra and its Applications 415 (1) (2006) 20 – 30. `doi:10.1016/j.laa.2005.07.021`.

[92] H. Zha, H. D. Simon, On updating problems in latent semantic indexing, SIAM Journal on Scientific Computing 21 (2) (1999) 782–791. `doi:10.1137/S1064827597329266`.

[93] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, in: Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition.

[94] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324. `doi:10.1109/5.726791`.

[95] P. Howland, J. Wang, H. Park, Solving the small sample size problem in face recognition using generalized discriminant analysis, Pattern Recognition 39 (2) (2006) 277–287. `doi:10.1016/j.patcog.2005.06.013`.

[96] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720. `doi:10.1109/34.598228`.

[97] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data with application to face recognition, Pattern Recognition 34 (2001) 2067–2070. `doi:10.1016/S0031-3203(00)00162-X`.

[98] L.-F. Chen, H.-Y. Liao, M.-T. Ko, J.-C. Lin, G.-J. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (10) (2000) 1713–1726. `doi:10.1016/S0031-3203(99)00139-9`.

[99] J. Yang, J. Yang, Why can LDA be performed in PCA transformed space?, Pattern Recognition 36 (2) (2003) 563–566. `doi:10.1016/S0031-3203(02)00048-1`.

[100] J. Ye, Q. Li, LDA/QR: an efficient and effective dimension reduction algorithm and its theoretical foundation, Pattern Recognition 37 (4) (2004) 851 – 854. `doi:10.1016/j.patcog.2003.08.006`.

[101] J. Ye, R. Janardan, C. Park, H. Park, An optimization criterion for generalized discriminant analysis on undersampled problems, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 982–994. `doi:10.1109/TPAMI.2004.37`.

[102] X. Wang, X. Tang, Dual-space linear discriminant analysis for face recognition, in: IEEE Conference on CVPR '04, Vol. 2, 2004, pp. 564–569. `doi:10.1109/CVPR.2004.1315214`.

[103] M. Zhu, A. M. Martinez, Subclass discriminant analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (8) (2006) 1274–1286. `doi:10.1109/TPAMI.2006.172`.

[104] J. Ye, Least squares linear discriminant analysis, in: 24th International Conference on Machine Learning, 2007, pp. 1087–1093. `doi:10.1145/1273496.1273633`.

[105] C. C. Paige, M. A. Saunders, Lsqr: An algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Softw. 8 (1) (1982) 43–71. `doi:10.1145/355984.355989`.

[106] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, Discriminative common vectors for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 27 (1) (2005) 4–13. `doi:10.1109/TPAMI.2005.9`.

[107] A. Tamura, Q. Zhao, Rough common vector: A new approach to face recognition, in: 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 2366–2371. `doi:10.1109/ICSMC.2007.4413825`.

[108] H. Cardot, D. Degras, Online principal component analysis in high dimension: Which algorithm to choose?, International Statistical Review n/a–n/a`doi:10.1111/insr.12220`.