Covariance-free Partial Least Squares: An Incremental Dimensionality Reduction Method

Artur Jordao, Maiko Lie, Victor Hugo Cunha de Melo and William Robson Schwartz
Smart Sense Laboratory, Computer Science Department
Federal University of Minas Gerais, Brazil
Email: {arturjordao, maikolie, victorhcmelo, william}@dcc.ufmg.br

Abstract

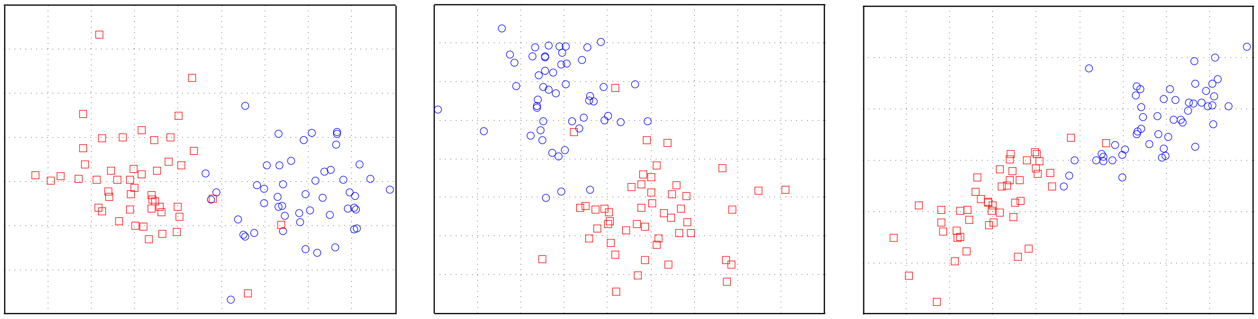
Dimensionality reduction plays an important role in computer vision problems since it reduces computational cost and is often capable of yielding more discriminative data representation. In this context, Partial Least Squares (PLS) has presented notable results in tasks such as image classification and neural network optimization. However, PLS is infeasible on large datasets, such as ImageNet, because it requires all the data to be in memory in advance, which is often impractical due to hardware limitations. Additionally, this requirement prevents us from employing PLS on streaming applications where the data are being continuously generated. Motivated by this, we propose a novel incremental PLS, named Covariance-free Incremental Partial Least Squares *(CIPLS), which learns a low-dimensional representation of the data using a single sample at a time. In contrast to other state-of-the-art approaches, instead of adopting a partially-discriminative or SGD-based model, we extend Nonlinear Iterative Partial Least Squares (NIPALS) — the standard algorithm used to compute PLS — for incremental processing. Among the advantages of this approach are the preservation of discriminative information across all components, the possibility of employing its score matrices for feature selection, and its computational efficiency. We validate CIPLS on face verification and image classification tasks, where it outperforms several other incremental dimensionality reduction techniques. In the context of feature selection, CIPLS achieves comparable results when compared to state-of-the-art techniques.*

1. Introduction

Dimensionality reduction is widely used in computer vision applications from image classification [11] [2] to detection of adversarial images [12]. The idea behind this technique is to estimate a transformation matrix that projects the high-dimensional feature space onto a low-dimensional latent space [23][8]. Previous works have demonstrated that dimensionality reduction can improve not only computational cost but also the effectiveness of the data representation [19] [35] [33]. In this context, Partial Least Squares (PLS) has presented remarkable results when compared to other dimensionality reduction methods [33]. This is mainly due to the criterion through which PLS finds the low dimensional space, which is by capturing the relationship between independent and dependent variables. Another interesting aspect of PLS is that it can operate as a feature selection method, for instance, by employing Variable Importance in Projection (VIP) [24]. The VIP technique employs score matrices yielded by NIPALS (the standard algorithm used for traditional PLS) to compute the importance of each feature based on its contribution to the generation of the latent space.

Despite achieving notable results, PLS is not suitable for large datasets, such as ImageNet [6], since it requires all the data to be in memory in advance, which is often impractical due to hardware limitations. Additionally, this requirement prevents us from employing PLS on streaming applications, where the data are being generated continuously. Such limitation is not particular to PLS, many dimensionality reduction methods, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), also suffer from this problem [36, 2, 39].

To handle the aforementioned problem, many works have proposed incremental versions of traditional dimensionality reduction methods. The idea behind these methods is to estimate the projection matrix using a single data sample (or a subset) at a time while keeping some properties of the traditional dimensionality reduction methods. A well-known class of incremental methods is the one based on Stochastic Gradient Descent (SGD) [3] [2]. These methods interpret dimensionality reduction as a stochastic optimization problem of an unknown distribution. As shown by Weng et al. [36], incremental methods based on SGD are computationally expensive, present convergence problems and require many parameters that depend on the nature of



(a) IPLS projection.

(b) SGDPLS projection.

(c) CIPLS (Ours) projection.

Figure 1. Projection on the first (x-axis) and second (y-axis) components using different dimensionality reduction techniques. Our method (CIPLS) separates the feature space better than IPLS and SGDPLS, which are state-of-the-art incremental PLS-based methods. For IPLS and SGDPLS, the class separability is effective only on a single dimension of the latent space, while for CIPLS it is retained on both dimensions. Blue and red points denote positive and negative samples, respectively.

the data. To address this problem, Zeng et al. [40] proposed an efficient and low-cost incremental PLS (IPLS). In their work, the first dimension (component) of the latent space is found incrementally, while the other dimensions are estimated by projecting the first component onto the reconstructed covariance matrix, which is employed to address the issue of impractical memory requirements of a full covariance matrix.

Even though IPLS achieves better performance than SGD-based and other state-of-the-art incremental methods, the discriminability of its higher-order components (i.e., all except the first) is not preserved, as shown in Figure 1 (a), where it can be seen that the effectiveness of class separability of IPLS is restricted to the first dimension of the latent space. This behavior occurs because the higher-order components are estimated using only the independent variables, that is, they are based on an approximation of the covariance matrix $X^\top X$ (similar to PCA) instead of $X^\top Y$ employed in PLS. This can degrade the discriminability of the latent model since preserving the relationship between independent and dependent variables is an important property of the original PLS [8]. It is important to emphasize that, for high-dimensional data, employing several components often provides better results [33, 9, 10], hence, IPLS might not be suitable for these cases.

Motivated by limitations and drawbacks in incremental PLS-based approaches, we propose a novel incremental method[1]. Our method is based on the hypothesis that the estimation of higher-order components using the covariance matrix, as proposed by Zeng et al. [40], is inadequate since the relationship between independent and dependent variables is lost. Therefore, to preserve this characteristic, we extend NIPALS [1] to avoid the computation of $X^\top Y$ and, consequently, enable it for incremental operation. Since our proposed extension is based on a simple algebraic decomposition, we preserve the simplicity and efficiency that makes NIPALS attractive, and we ensure that the relationship between independent and dependent variables is propagated to all components, differently from other methods.

As shown in Figure 1, our method is capable of separating data classes better than IPLS, mainly on the second component (i.e., y-axis). Since the proposed method does not use the covariance matrix $(X^\top X)$ to estimate higher-order components, we refer to it as *Covariance-free Incremental Partial Least Squares* (CIPLS). Besides providing superior performance, our method can easily be extended as a feature selection technique since it provides all the requirements to perform VIP. Existing incremental PLS methods, on the other hand, require more complex techniques to operate as feature selection [24].

We compare the proposed method on the tasks of face verification and image classification, where it outperforms several other incremental methods in terms of accuracy and efficiency. In addition, in the context of feature selection, we evaluate and compare the proposed method to state-of-the-art methods, where it achieves competitive results.

2. Related Work

To enable PCA to operate in an incremental scheme, Weng et al. [36] proposed to compute the principal components without estimating the covariance matrix, which is unknown and impossible to be calculated in incremental methods. For this purpose, their method, named CCIPCA, updates the projection matrix for each sample x, replacing the unknown covariance matrix by the sample covariance matrix $xx^\top$. While CCIPCA provides a minimum reconstruction error of the data, it might not improve or even preserve the discriminability of the resulting subspace since label information is ignored (similarly to traditional PCA) [23].

To achieve discriminability, incremental methods based on Linear Discriminant Analysis (LDA) have been pro-

[1] https://github.com/arturjordao/IncrementalDimensionalityReduction

posed [13] [21]. In particular, this class of methods is less explored since they present issues such as the *sample size problem* [14], which makes them infeasible for some tasks. Different from incremental LDA methods, incremental PLS methods are more flexible and present better results [40]. Motivated by this, Arora et al. [3] proposed an incremental PLS based on stochastic optimization (SGDPLS), where the idea is to optimize an objective function using a single sample at a time. Similarly to Arora et al. [3], Stott et al. [34] proposed applying stochastic gradient maximization on NIPALS, extending it for incremental processing. Even though they present promising results on synthetic data, their approach presented convergence problems when evaluated on real-world datasets. Thus, in this work, we consider only the approach by Arora et al. [3], which was the one that converged for several of the datasets evaluated and presented better results.

While SGDPLS is effective, as demonstrated by Weng et al. [36] and Zeng et al. [40], SGD-based methods applied to dimensionality reduction are computationally expensive and present convergence problems. In addition, this class of approaches requires careful parameter tuning and their results are often sensitive to the type of dataset [36].

To address convergence problems in SGD-based PLS, Zeng et al. [40] proposed to decompose the relationship between independent and dependent matrices (variables) into a sample relationship (i.e., a single sample with its label). This process is performed only to compute the first component, while the higher-order components are estimated by projecting the first component onto an approximated covariance matrix using a few PCA components. As we mentioned earlier, since traditional PCA cannot be employed in incremental methods, Zeng et al. [40] used CCIPCA to reconstruct the principal components of the covariance matrix.

In contrast to existing incremental PLS methods, our method presents superior performance in both accuracy and execution time for estimation of the projection matrix, which is an important requirement for time-sensitive and resource-constrained tasks. In particular, considering the average accuracy across all tasks in our assessment, the proposed method outperforms IPLS and SGDPLS by 32.48 and 24.83 percentage points, respectively, when using only higher-order components. The reason for these results is the quality of our higher-order components, which keeps the discriminative properties of traditional PLS.

Another line of research widely employed to reduce computational cost is feature selection. One of the most successful feature selection methods is the work by Roffo et al. [32], which proposed to interpret feature selection as a graph problem. In their method, named *infinity feature selection* (infFS), each feature represents a node in an undirected fully-connected graph and the paths in this graph represent the combinations of features. Following this model, the goal is to find the best path taking into account all the possible paths (in this sense, all the subsets of features) on the graph, by exploring the convergence property of the geometric power series of a matrix. Improving upon this model, Roffo et al. [30] suggested quantizing the raw features into a small set of tokens before computing infFS. By using this pre-processing, their method (referred to as *infinity latent feature selection* — ilFS) achieved even better results than infFS. Recently, Roffo et al. [31] presented a more efficient version of infFS, which considers supervised (infFS_S) and unsupervised (infFS_U) scenarios. Although the framework by Roffo et al. [32, 30, 31] achieved state-of-the-art results, in the context of neural network optimization, Jordao et al. [17] showed that PLS+VIP attains superior performance. We show that CIPLS+VIP achieves comparable results when compared to PLS+VIP and other state-of-the-art feature selection techniques.

3. Proposed Approach

In this section, we start by describing the traditional Partial Least Squares (PLS). Then, we present the proposed *Covariance-free Incremental Partial Least Squares* (CIPLS) and the Variable Importance in Projection (VIP) technique, which enables PLS and CIPLS to be employed for feature selection. Unless stated otherwise, let $X \in \mathbb{R}^{n \times m}$ be the matrix of independent variables denoting n training samples in a m-dimensional space. Furthermore, let $Y \in \mathbb{R}^{n \times 1}$ be the matrix of dependent variables representing the binary class label. Finally, let $x_n \in \mathbb{R}^{1 \times m}$ and $y_n \in \mathbb{R}^{1 \times 1}$ be a single sample of X and Y, respectively. We highlight that, in the context of streaming data, x_n is a data sample acquired at time n.

3.1. Partial Least Squares

Given a high m-dimensional space, PLS finds a projection matrix $W(w_1, w_2, ..., w_c)$, which projects this space onto a low c-dimensional space, where $c \ll m$. For this purpose, PLS aims at maximizing the covariance between the independent and dependent variables such that, besides reducing dimensionality, it preserves the discriminability of the data, which is essential for classification tasks. Formally, PLS constructs W such that

$$w_i = maximize(\text{cov}(Xw, Y)), \text{s.t}\|w\| = 1, \quad (1)$$

where w_i denotes the ith component of the c-dimensional space. The exact solution to Equation 1 is given by

$$w_i = \frac{X^\top Y}{\|X^\top Y\|}. \quad (2)$$

From Equation 2, we can compute all the c components using either Nonlinear Iterative Partial Least Squares (NIPALS) [1] or Singular Value Decomposition (SVD). Most works employ NIPALS since it is capable of finding only the c first components, while SVD always finds all the m components, being computationally prohibitive for large datasets [38, 22].

3.2. Covariance-free Incremental Partial Least Squares

The core idea in our method is to ensure that, as in traditional PLS, the relationship between independent and dependent variables (Equation 2) is kept on all the c components. To achieve this goal, our method works as follows. First, we need to center the data to the mean of the training samples X. However, different from traditional methods, in incremental approaches the mean is unknown since we cannot assume that all the data are known a priori [36] [40]. To face this problem, we center the current data sample using an approximate centralization process [36] which consists of estimating an incremental mean using the nth sample. According to Weng et al. [36], we can compute the incremental mean μ_n w.r.t. the nth data sample as

$$\mu_n = \frac{n-1}{n}\mu_{(n-1)} + \frac{1}{n}x_n. \tag{3}$$

Once we have centralized the sample, the next step in our method is to compute the component w_i following Equation 2. As we mentioned, X and its respective Y are unknown or are not in memory in advance, which prohibits us to apply Equation 2 directly. However, as suggested by Zeng et al. [40], we employ the following decomposition:

$$X^T Y = \sum_{k=1}^{n-1} x_k^T y_k + x_n^T y_n. \tag{4}$$

By replacing $X^\top Y$ in Equation 2 by Equation 4, it is possible to calculate the ith component of PLS considering a single sample at a time. In other words, Equation 4 enables to compute w_i incrementally.

To compute the higher-order components ($w_i, i > 1$), we employ a *deflation* process that consists of subtracting the contribution of the current component on the sample before estimating the next component. Following the NIPALS algorithm, the deflation process works as follows

$$t = Xw_i, \tag{5}$$

$$p = X^\top t, q = Y^\top t, \tag{6}$$

$$X = X - tp^\top, Y = Y - tq^\top, \tag{7}$$

where t denotes the projected samples onto the current component w_i, and p and q represent the loadings of this projection. It should be noted that while t works in an incremental scheme (since we can project one sample at a time), p and q cannot be computed since X and Y are neither known nor are in memory in advance. However, in light of Equation 4, we can decompose p and q as

$$p = \sum_{k=1}^{n-1} x_k^\top t_k + x_n^\top t_n, q = \sum_{k=1}^{n-1} y_k^\top t_k + y_n^\top t_n. \tag{8}$$

By embedding Equation 8 in the deflation process, we can remove the contribution of the current component and repeat the process to compute a single component w_i (as we argued before). Observe that Equation 7 deflates each sample by its reconstructed value. This way, Equation 7 can be computed sample-by-sample, working in an incremental scheme. With this formulation, we are now capable of computing the c components incrementally. Algorithm 1 summarizes the steps of our method. It should be mentioned that the matrices W, P and Q are initialized with zeros.

According to Algorithm 1, the proposed method maintains the property of capturing the relationship between X and Y for all components (step 4 in Algorithm 1). In addition, since we compute all components at once, our method has a time complexity of $O(ncm)$, where n, c and m denote the number of samples, number of components, and dimensionality of the data, respectively.

Algorithm 1: CIPLS Algorithm.

```
Input  : $n$th data sample $x_n$ and its label $y_n$
         Number of components $c$
         Projection matrix $W_{(n-1)} \in \mathbb{R}^{m \times c}$
         Loading matrix $P_{(n-1)} \in \mathbb{R}^{m \times c}$
         Loading matrix $Q_{(n-1)} \in \mathbb{R}^{1 \times c}$
Output: Updated matrices $W$, $P$ and $Q$

1  Update $\mu_n$ using Equation 3
2  $\bar{x}_n = x_n - \mu_n$
3  for $i = 1$ to $c$ do
4      $w_i = \bar{x}_n^\top y_n + w_{i(n-1)}$, where $w_i \in W$
5      $t_n = \frac{\bar{x}_n w_i}{||\bar{x}_n w_i||}$
6      $p_i = \bar{x}_n^\top t_n + p_{i(n-1)}$, where $p_i \in P$
7      $q_i = y_n^\top t_n + q_{i(n-1)}$, where $q_i \in Q$
8      $\bar{x}_n = \bar{x}_n - t_n p_i^\top$
9      $y_n = y_n - t_n q_i^\top$
10 end
```

3.3. CIPLS for Feature Selection

An advantage of PLS is that, after estimating the projection matrix W, it is possible to estimate the importance of each feature, enabling PLS to operate as a feature selection method. For this purpose, it is possible to employ Variable

Importance in Projection (VIP), which estimates the importance of each feature f_j based on its contribution to yield the low dimensional space. According to Mehmood et al. [24], VIP is defined as

$$f_j = \sqrt{m \sum_{i=1}^{c} q_i^2 t_i^\top t_i (w_{ij}/\|w_i\|^2) / \sum_{i=1}^{c} q_i^2 t_i^\top t_i}. \quad (9)$$

Once we have estimated the score of each feature, we can remove a percentage of features based on their scores. As can be verified in Algorithm 1, CIPLS preserves the ability of traditional PLS to be employed as a feature selection method via VIP (Equation 9). It is important to emphasize that IPLS and SGDPLS cannot be used to compute VIP as they do not provide the loading matrix Q $(q_1, q_2, ..., q_c)$.

4. Experimental Results

In this section, we first introduce the experimental setup and the tasks employed to validate the proposed method. Then, we present the procedure conducted to calibrate the parameters of the methods. Next, we compare the proposed method with other incremental partial least squares methods as well as with the traditional PLS. Afterwards, we present the influence of higher-order components on the classification performance. Finally, we discuss the time complexity of the methods, their performance on a streaming scenario and compare our method on the feature selection context.

Experimental Setup. Following previous works [7, 28, 4, 18], we employ linear classifiers when using features from convolutional networks. Specifically, we use a linear SVM as employed by our main baseline [40]. For multi-class problems (image classification), on the other hand, we prefer to use a multilayer perceptron since it handles the multi-class problem naturally, avoiding the need for employing a binary classifier (e.g., SVM) on a one-versus-rest fashion, which would be computationally expensive. All experiments and methods were executed on an Intel Core i5-8400, 2.4 GHz processor with 16 GB of RAM.

To assess the differences in efficacy and efficiency among the compared methods, throughout the experiments we perform statistical tests based on a paired t-test using 95% confidence [16]. We highlight that the statistical tests were conducted only for face verification due to the computational cost of retraining (i.e., fine-tuning) the convolutional network for image classification, which is considerably high since we employ large-scale datasets.

Face Verification. Given a pair of face images, face verification determines whether this pair belongs to the same person. For this purpose, we use a three-stage pipeline [27, 5] as follows. First, we extract a feature vector of each face using a deep learning model. In this work, we use the feature maps from the last convolutional layer of the VGG16 model, learned on the VGGFaces dataset [26], as feature vector. Then, we compute the distance between the two feature vectors employing the ℓ_1-distance metric. Finally, we present the result of the distance metric to a classifier.

We conduct our evaluation on two face verification datasets, namely Labeled Faces in the Wild (LFW) [15] and Youtube Faces (YTF) [37].

Image Classification. For this task, we use features maps from the last layer of a VGG16 network as features. Moreover, we consider two versions of the ImageNet dataset, with images of size 224×224 and 32×32 pixels. The former is used since it is the original version of the dataset, while the latter is used because it has been demonstrated to be more challenging than the original version [29, 20, 25]. It is worth mentioning that the single difference between these versions of ImageNet is the image size.

Number of Components. One of the most important aspects of dimensionality reduction methods is the number of components c of the resulting latent space. Therefore, to choose the best number of components for each method, we vary c from 1 to 10 and select the value for which the method achieved the highest accuracy on the validation set (10% of the training set). Once the best c is chosen, we use the training and validation set to learn the projection method and the classifier. We repeat this process for each dataset.

Comparison with Incremental Methods. This experiment compares the proposed CIPLS with other incremental dimensionality reduction methods. Table 1 summarizes the results and shows that, on LFW, our method outperformed SGDPLS and IPLS by 1.18 and 1.48 percentage points (p.p.), respectively. Similarly, on YTF, CIPLS outperformed SGDPLS and IPLS by 0.88 and 1.88 p.p..

Finally, on the ImageNet dataset, the difference in accuracy compared to IPLS was of 0.07 and 1.35 p.p., for the 32×32 and 224×224 versions, respectively. It is important to mention that we do not consider SGDPLS on these datasets due to convergence problems and high computational cost. Furthermore, due to memory constraints, it was not possible to run the traditional PLS on ImageNet.

Comparison with Partial Least Squares. As suggested by Weng et al. [36], we compare the incremental methods with the traditional approach as baseline (in our case, traditional PLS). According to Table 1, besides providing better results than IPLS and SGDPLS, CIPLS achieved the closest results to traditional PLS. For instance, on LFW, the difference in accuracy between PLS and CIPLS was 0.69 p.p. while on YTF it was 1.86 p.p.. In contrast, the difference in accuracy between PLS and SGDPLS is higher — 1.87 p.p. on LFW and 2.74 p.p. on YTF. In addition, the difference in accuracy between PLS and IPLS is among the highest, 2.17 and 3.74 p.p. for the LFW and YTF datasets, respectively. In particular, the results for PLS and CIPLS are statistically

Table 1. Comparison of existing incremental methods in terms of accuracy. The symbol '–' denotes that it was not possible to execute the method on the respective dataset due to memory constraints or convergence problems (see the text). PLS denotes the use of the traditional PLS. The closer to the accuracy of the baseline (PLS), the better. The numbers enclosed in square brackets denote confidence interval (95% confidence).

	LFW	YTF	ImageNet	ImageNet
			32×32	224×224
CCIPCA [36]	89.87 $[89.17, 90.55]$	81.48 $[80.07, 82.88]$	40.30	52.58
SGDPLS [3]	90.60 $[89.95, 91.24]$	83.22 $[82.07, 84.36]$	–	–
IPLS [40]	90.30 $[89.60, 90.99]$	82.22 $[80.96, 83.47]$	43.24	65.74
CIPLS (Ours)	91.78 $[91.08, 92.47]$	84.10 $[82.82, 85.37]$	43.31	67.09
PLS	92.47 $[91.87, 93.05]$	85.96 $[84.47, 87.44]$	–	–

Table 2. Accuracy of existing incremental methods when using only higher-order components. Values computed considering the average accuracy across all tasks in our assessment.

	Average Accuracy
CCIPCA [36]	63.48
SGDPLS [3]	58.41
IPLS [40]	50.76
CIPLS (Ours)	83.24

equivalent, while IPLS and SGDPLS present results statistically inferior compared to PLS.

It should be noted that the results of IPLS are closer to CCIPCA than PLS since only the first component of IPLS maintains the relationship between independent and dependent variables. On the other hand, the proposed method preserves this relation along higher-order components, which provides better discriminability, as seen in our results.

Higher-order Components. This experiment assesses the discriminability of the higher-order components of CIPLS compared to each of the other incremental methods. For this purpose, we follow a process suggested by Martinez [23], which consists of removing the first component of the latent space before presenting the projected data to the classifier. This evaluates the performance of the remaining components, not only the first one which tends to be better.

Table 2 shows the results. According to Table 2, the proposed method outperforms IPLS by 32.48 p.p.. Observe that when all the components are used, CIPLS outperforms IPLS by 1.17 p.p.. This larger difference when removing the first component is an effect of the better discriminability achieved by the components extracted by CIPLS. As we have argued, CIPLS preserves the relationship between dependent and independent variables across higher-order components, yielding more accurate results. Compared to SGDPLS, CIPLS outperforms it by 24.83 p.p. when using only the higher-order components.

Table 3. Comparison of incremental dimensionality reduction methods in terms of time complexity for estimating the projection matrix. m, n denote dimensionality of the original data and number of samples, while c, L and T denote number of PLS components, number of PCA components and convergence steps.

	Time Complexity
CCIPCA [36]	$O(nLm)$
SGDPLS [3]	$O(Tcm)$
IPLS [40]	$O(nLm + c^2m)$
CIPLS (Ours)	$O(ncm)$

Time Issues. To demonstrate the efficiency of CIPLS, in this experiment, we compare its time complexity to compute the projection matrix with the incremental methods evaluated. Following Weng et al. [36] and Zeng et al. [40], we report this complexity w.r.t. dimensionality of the original data m, number of samples n, number of components c and number of PCA components L (required only by IPLS and CCIPCA). Table 3 shows the time complexity of the methods.

According to Table 3, CIPLS presents a low time com-

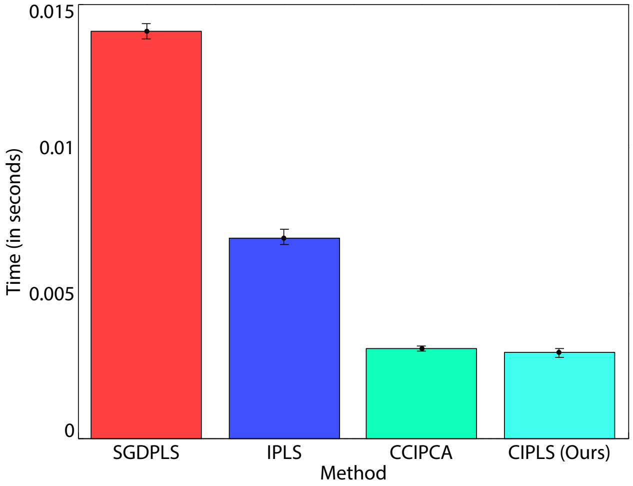
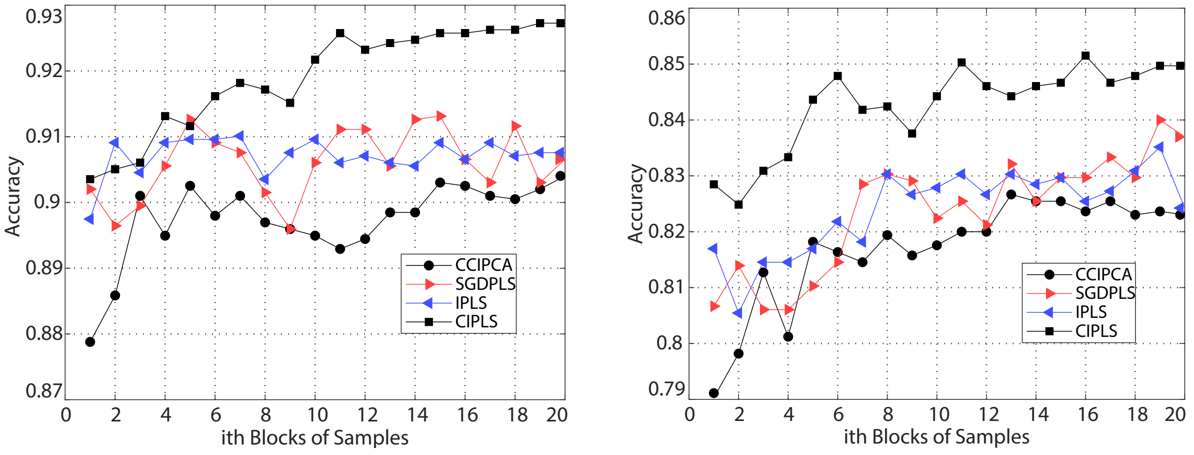


Figure 2. Average prediction time (in seconds) for estimating the projection matrix, lower values are better. Black bars denote the confidence interval.



(a) Labeled Faces in the Wild (LFW).

(b) Youtube Faces (YTF).

Figure 3. Comparison of incremental methods on a streaming scenario. The x-axis denotes the data arriving sequentially.

plexity for estimating the projection matrix. The complexity of CIPLS is not only on the same class as CCIPCA, which is the fastest among the compared methods, but it also has a very small constant factor. This constant factor is the number of components, c for CIPLS and L for CCIPCA. Experimentally, we found that the optimal constant factor for PLS is negligible, $c = 2$ resulted in the highest accuracies. While, for fairness, the same number of components was adopted for all methods in Table 3, typically $c < L$ on practical applications. This is a known advantage of PLS – it has been shown to require substantially less components to achieve its optimal accuracy than PCA [33].

We also report the average computation time (considering 30 executions) of the methods for estimating the projection matrix for one new sample. To make a fair comparison, we set $c = 4$ for all methods and for the other parameters we use the values where the methods achieved the best results in validation. According to Figure 2, SGDPLS is the slowest incremental PLS method, which is a consequence of its strategy for estimating the projection matrix, where for each sample the convergence step is run T times. Our experiments showed that $T \geq 100$ is required for good results. The computation time for estimating the projection matrix of our method was statistically equivalent (according to a paired t-test) to that of CCIPCA, which is the fastest among the incremental dimensionality reduction methods assessed. Moreover, CIPLS was statistically faster than IPLS and SGDPLS, demonstrating that it is the fastest among the compared incremental PLS methods.

Incremental Methods on the Streaming Scenario. As we argued earlier, incremental methods can be employed on streaming applications, where the training data are continuously generated. To demonstrate the robustness of our method on these scenarios, we evaluate the methods on a synthetic streaming context, as proposed by Zeng et al. [40]. The procedure works as follows. First, the training data is divided into k blocks, where $k = 20$. The idea behind this process is to interpret each block as a new instance of arriving data. Then, we create a new training set and insert each kth block at a time. Each time we insert a new block, we learn the projection method and evaluate its accuracy on the testing set. For instance, when adding the tenth block, all the $1, 2, ..., 10$ blocks are being used as training. It is important to mention that a block contains more than one sample, however, this does not modify the strategy of the incremental methods, which is to estimate the projection matrix by using a single sample at a time.

Figure 3 (a) and (b) show the results on the LFW and YTF datasets, respectively. On the LFW dataset, until the fifth block, it is not possible to determine the best method since the accuracy presents high variance, however, from the sixth block onwards, CIPLS outperforms all other methods. On the YTF dataset, our method achieves the highest accuracy for all blocks. These results show that the proposed method is more adequate for streaming applications than existing incremental PLS methods.

Comparison with Feature Selection Methods. Our last experiment evaluates the performance of CIPLS as a feature selection method. Table 4 shows the results for different percentages of kept features on the LFW and YTF datasets.

According to Table 4, CIPLS is on par with the state-of-the-art feature selection techniques. For example, on LFW the difference in accuracy, on average, from CIPLS to infFS and ilFS is of 0.15 and 0.25 p.p., respectively. Compared to infFS$_\mathrm{S}$ and infFS$_\mathrm{U}$, this difference is 0.05 and 0.26 p.p., in this order. Interestingly, on YTF for some percentages of kept features (e.g., 15% and 50%), CIPLS outperforms ilFS, infFS$_\mathrm{S}$ and infFS$_\mathrm{U}$. We highlight that these methods were designed specifically for feature selection.

Finally, the difference, on average, between CIPLS and

Table 4. Comparison of feature selection methods using different percentages of kept features.

	LFW				YTF			
	Percentage of Kept Features				Percentage of Kept Features			
	10	15	20	50	10	15	20	50
infFS [32]	91.92	91.58	92.03	92.23	86.64	86.68	87.14	87.30
ilFS [30]	92.03	91.67	92.25	92.23	86.60	86.94	86.84	87.54
infFS_U [31]	92.08	91.70	92.30	92.15	86.36	86.60	87.14	87.16
infFS_S [31]	91.80	91.62	91.62	92.33	86.12	86.50	86.80	87.22
PLS+VIP	92.05	91.67	92.13	92.38	86.70	86.82	87.18	87.68
CIPLS (Ours)+VIP	91.63	91.55	91.80	92.18	86.48	86.92	87.02	87.40

PLS is of 0.26 and 0.14 p.p. on the LFW and YTF datasets. Moreover, the largest accuracy difference between PLS and CIPLS is only 0.4 p.p., on LFW with 10% of features kept. This result reinforces that the proposed decompositions to extend the NIPALS and enable the employment of VIP are a good approximation of the original method.

Based on the results shown, it is possible to conclude that, besides dimensionality reduction, the proposed method achieves state-of-the-art results in the context of feature selection.

5. Conclusions

This work presented a novel incremental partial least squares method, named *Covariance-free Incremental Partial Least Squares* (CIPLS). The method extends the NIPALS algorithm for incremental operation and enables computation of the projection matrix using one sample at a time while still presenting the main property of traditional PLS, namely preserving the relation between dependent and independent variables. Compared to existing incremental partial least squares methods, CIPLS achieves superior performance besides being computationally efficient. In addition, different from previous incremental partial least squares, CIPLS can easily to operate as a feature selection method. In this context, the proposed method is able to achieve comparable results to the state of the art.

Acknowledgments

The authors would like to thank the National Council for Scientific and Technological Development – CNPq (Grants 140082/2017-4, 438629/2018-3 and 309953/2019-7) and the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17.

References

[1] Hervé Abdi. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2010.

[2] Salaheddin Alakkar and John Dingliana. An acceleration scheme for mini-batch, streaming pca. *BMVC*, 2019.

[3] Raman Arora, Poorya Mianjy, and Teodor V. Marinov. Stochastic optimization for multiview representation learning using partial least squares. In *ICML*, 2016.

[4] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[5] Jun-Cheng Chen, Rajeev Ranjan, Swami Sankaranarayanan, Amit Kumar, Ching-Hui Chen, Vishal M. Patel, Carlos D. Castillo, and Rama Chellappa. Unconstrained still/video-based face verification with deep convolutional neural networks. *IJCV*, 2018.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, pages 248–255, 2009.

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[8] Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 1986.

[9] Guodong Guo and Guowang Mu. Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression. In *CVPR*, 2011.

[10] Guodong Guo and Guowang Mu. Joint estimation of age, gender and ethnicity: CCA vs. PLS. In *FG*, 2013.

[11] Ryoma Hasegawa and Kazuhiro Hotta. Plsnet: A simple network using partial least squares regression for image classification. In *ICPR*, 2016.

[12] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. In *ICLR*, 2017.

[13] Kazuyuki Hiraoka, Shuji Yoshizawa, Ken-ichi Hidai, Masashi Hamahira, Hiroshi Mizoguchi, and Taketoshi

Mishima. Convergence analysis of online linear discriminant analysis. In *IJCNN*, pages 387–391, 2000.

[14] Peg Howland, Jianlin Wang, and Haesun Park. Solving the small sample size problem in face recognition using generalized discriminant analysis. *Pattern Recognition*, 2006.

[15] Gary B. Huang, Marwan A. Mattar, Honglak Lee, and Erik G. Learned-Miller. Learning to align from scratch. In *NeurIPS*, pages 773–781, 2012.

[16] Raj Jain. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley professional computing. John Wiley & Sons, 1990.

[17] Artur Jordao, Maiko Lie, and William Robson Schwartz. Discriminative layer pruning for convolutional neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[18] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *CVPR*, 2019.

[19] Marc T. Law, Jake Snell, Amir-massoud Farahmand, Raquel Urtasun, and Richard S. Zemel. Dimensionality reduction for representing the knowledge of probabilistic models. In *ICLR*, 2019.

[20] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[21] Gui-Fu Lu, Jian Zou, and Yong Wang. Incremental learning of complete linear discriminant analysis for face recognition. *Knowledge-Based Systems*, 2012.

[22] Alaa Maalouf, Ibrahim Jubran, and Dan Feldman. Fast and accurate least-mean-squares solvers. In *NeurIPS*, 2019.

[23] Aleix M. Martínez and Avinash C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[24] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 2012.

[25] Zoltan A. Milacski, Barnabas Poczos, and Andras Lorincz. Differentiable unrolled alternating direction method of multipliers for onenet. *BMVC*, 2019.

[26] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015.

[27] Rajeev Ranjan, Swami Sankaranarayanan, Ankan Bansal, Navaneeth Bodla, Jun-Cheng Chen, Vishal M. Patel, Carlos D. Castillo, and Rama Chellappa. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *Signal Processing Magazine*, 2018.

[28] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR*, 2014.

[29] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, 2017.

[30] Giorgio Roffo, Simone Melzi, Umberto Castellani, and Alessandro Vinciarelli. Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In *ICCV*, 2017.

[31] Giorgio Roffo, Simone Melzi, Umberto Castellani, Alessandro Vinciarelli, and Marco Cristani. Infinite feature selection: a graph-based feature filtering approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[32] Giorgio Roffo, Simone Melzi, and Marco Cristani. Infinite feature selection. In *ICCV*, 2015.

[33] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human detection using partial least squares analysis. In *ICCV*, pages 24–31, 2009.

[34] Alexander E. Stott, Sithan Kanna, Danilo P. Mandic, and William T. Pike. An online NIPALS algorithm for partial least squares. In *ICASSP*, pages 4177–4181, 2017.

[35] Bing Su and Ying Wu. Learning low-dimensional temporal representations. In *ICML*, 2018.

[36] Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[37] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011.

[38] Zhiqiang Xu and Ping Li. Towards practical alternating least-squares for CCA. In *NeurIPS*, 2019.

[39] Le Yang, Shiji Song, Yanshang Gong, Huang Gao, and Cheng Wu. Nonparametric dimension reduction via maximizing pairwise separation probability. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[40] Xue-Qiang Zeng and Guo-Zheng Li. Incremental partial least squares analysis of big streaming data. *Pattern Recognition*, 47:3726–3735, 2014.