

AN INTEGRATED ALGORITHM OF INCREMENTAL AND ROBUST PCA

Yongmin Li, Li-Qun Xu, Jason Morphet and Richard Jacobs

Content and Coding Lab, BTexact Technologies, Adastral Park, Ipswich IP5 3RE, UK
Email: {Yongmin.Li, Li-Qun.Xu, Jason.Morphett, Richard.J.Jacobs}@bt.com

ABSTRACT

Principal Component Analysis (PCA) is a well-established technique in image processing and pattern recognition. Incremental PCA and robust PCA are two interesting problems with numerous potential applications. However, these two issues have only been separately addressed in the previous studies. In this paper, we present a novel algorithm for incremental and robust PCA by seamlessly integrating the two issues together. The proposed algorithm has the advantages of both incremental PCA and robust PCA. Moreover, unlike most M-estimation based robust algorithms, it is computationally efficient. Experimental results on dynamic background modelling are provided to show the performance of the algorithm with a comparison to the conventional batch-mode and non-robust algorithms.

1. INTRODUCTION

Principal Component Analysis (PCA) has been extensively applied to numerous applications in image processing and pattern recognition. While it can efficiently represent high dimensional vectors with a small number of orthogonal basis vectors, the conventional methods of PCA usually perform in batch-mode which is computationally expensive when dealing with large scale problems. To address this problem, there have been several *incremental* algorithms developed in the previous studies [1, 2, 3, 4]. These algorithms are generally similar in terms of accuracy and speed while the differences are mainly on how to approximate the covariance matrix.

In addition, the traditional PCA, in the sense of least mean squared error minimisation, is susceptible to outlying measurements. To enable PCA less vulnerable to "outliers", it has been proposed to use robust methods for PCA computation [5, 6, 7]. Unfortunately most of these methods are computationally intensive because the optimisation problem has to be computed *iteratively*, e.g. the self-organising algorithms in [5], the criss-cross regressions in [6] and the Expectation Maximisation algorithm in [7].

To address the two problems discussed above, we present an integrated algorithm for incremental and robust PCA computing. To our knowledge, these two issues have not yet been addressed together in the previous studies. We will give the detailed mathematical derivation of the algorithm and demonstrate its computational efficiency and robustness with experimental results on background modelling.

2. INCREMENTAL AND ROBUST PCA LEARNING

In the next two sub-sections, we will present the mathematical derivation of our algorithm for incremental and robust PCA before

the abstract algorithm description in Section 2.3.

2.1. Incremental PCA Updating

PCA seeks to compute the optimal linear transform with reduced dimensionality in the sense of least mean squared reconstruction error. This problem is usually solved using batch-mode algorithms such as eigenvalue decomposition and Singular Value Decomposition which are computationally intensive when applied to large scale problems where both the dimensionality and number of training examples are large. Incremental algorithms can be employed to provide approximate solutions with simplified computation. Along with the algorithms presented in the previous work [1, 2, 3, 4], we present a novel incremental algorithm in this paper.

Given a new observation vector x which has been subtracted by the mean vector μ , i.e.

$$x = x' - \mu \quad (1)$$

where x' is the original observation vector, if we assume the updating weights on the previous PCA model and the current observation vector are α and $1 - \alpha$ respectively, the mean vector can be updated as

$$\mu^{new} = \alpha\mu + (1 - \alpha)x' = \mu + (1 - \alpha)x \quad (2)$$

Construct $p + 1$ vectors from the previous eigenvectors and the current observation vector

$$y_i = \sqrt{\alpha\lambda_i}u_i, \quad i = 1, 2, \dots, p \quad (3)$$

$$y_{p+1} = \sqrt{1 - \alpha}x \quad (4)$$

where $\{u_i\}$ and $\{\lambda_i\}$ are the current eigenvectors and eigenvalues. The PCA updating problem can then be solved as an eigen-decomposition problem on the $p + 1$ vectors. An $n \times (p + 1)$ matrix A can then be defined as

$$A = [y_1, y_2, \dots, y_{p+1}] \quad (5)$$

Assume the covariance matrix C can be approximated by the first p significant eigenvectors and their corresponding eigenvalues,

$$C \approx U_{np}\Lambda_{pp}U_{np}^T \quad (6)$$

where the columns of U s are eigenvectors of C and diagonal matrices Λ s are comprised of eigenvalues of C . With a new observation x , the new covariance matrix is expressed by

$$\begin{aligned} C^{new} &= \alpha C + (1 - \alpha)xx^T \\ &\approx \alpha U_{np}\Lambda_{pp}U_{np}^T + (1 - \alpha)xx^T \\ &= \sum_{i=1}^p \alpha \lambda_i u_i u_i^T + (1 - \alpha)xx^T \end{aligned} \quad (7)$$

Substituting (3), (4) and (5) into (7) gives

$$C^{new} = AA^T \quad (8)$$

Instead of the $n \times n$ matrix C^{new} , we eigen-decompose a smaller $(p+1) \times (p+1)$ matrix B ,

$$B = A^T A \quad (9)$$

yielding eigenvectors $\{v_i^{new}\}$ and eigenvalues $\{\lambda_i^{new}\}$ which satisfy

$$A^T A v_i^{new} = \lambda_i^{new} v_i^{new}, \quad i = 1, 2, \dots, p+1 \quad (10)$$

Left multiplying by A on both sides, we have

$$A A^T A v_i^{new} = \lambda_i^{new} A v_i^{new} \quad (11)$$

Defining

$$u_i^{new} = A v_i^{new} \quad (12)$$

and then using (8) and (12) in (11) leads to

$$C^{new} u_i^{new} = \lambda_i^{new} u_i^{new} \quad (13)$$

i.e. u_i^{new} is an eigenvector of C^{new} with eigenvalue λ_i^{new} .

Note that the proposed algorithm performs similarly to other algorithms in terms of accuracy and complexity. The difference to other algorithms is how to express the covariance matrix incrementally (Equation (7)). Thus, it at least provides an alternative solution to the problem; and moreover, our algorithm can be interpreted as approximating the original large scale PCA with a small scale PCA on $\{y_1, y_2, \dots, y_{p+1}\}$, which is more concise and analytical in the way of presentation. It is also important to note that, like other incremental algorithms, the update rate α , which determines the weights on the previous information and new information, is application-dependent and has to be chosen experimentally.

2.2. Robust Analysis

Recall that PCA is the optimal linear transform in the sense of least squared reconstruction error. When the data used to construct the PCA contain contaminated outlying measurement, the conventional PCA may deviate from the desired solution. As oppose to the iterative algorithms of robust PCA developed previously [5, 6, 7], we present a simplified method as follows.

Given a PCA model, the residual error of a vector x_i is expressed by

$$r_i = U_{np} U_{np}^T x_i - x_i \quad (14)$$

We know that the conventional non-robust PCA is the solution of a least-squares problem¹

$$\min \sum_i \|r_i\|^2 = \sum_i \sum_j (r_i^j)^2 \quad (15)$$

Instead of sum-of-squares, the robust M-estimation method [8] seeks to solve the following problem via a robust function $\rho(r)$

$$\min \sum_i \sum_j \rho(r_i^j) \quad (16)$$

Differentiating (16) by θ_k , the parameters to be estimated, i.e. the elements of U_{np} , we have

$$\sum_i \sum_j \psi(r_i^j) \frac{\partial r_i^j}{\partial \theta_k} = 0, \quad k = 1, 2, \dots, np \quad (17)$$

where $\psi(t) = d\rho(t)/dt$ is the influence function. By introducing a weight function

$$w(t) = \frac{\psi(t)}{t} \quad (18)$$

¹In this context, we use subscript to denote the index of vectors, and superscript the index of their elements.

Equation (17) can be written as

$$\sum_i \sum_j w(r_i^j) r_i^j \frac{\partial r_i^j}{\partial \theta_k} = 0, \quad k = 1, 2, \dots, np \quad (19)$$

which is exactly the solution of a new least-squares problem

$$\min \sum_i \sum_j w(r_i^j) (r_i^j)^2 \quad (20)$$

If we define

$$z_i^j = \sqrt{w(r_i^j)} x_i^j \quad (21)$$

then substituting (14) and (21) into (20) leads to a new eigen-decomposition problem

$$\min \sum_i \|U_{np} U_{np}^T z_i - z_i\|^2 \quad (22)$$

It is important to note that w is a function of the residual error r_i^j which needs to be computed for each individual training vector (subscript i) and each of its elements (superscript j). The former maintains the adaptability of the algorithm, while the latter ensures the algorithm is robust to every element of a sample vector.

If we choose the robust function as the Cauchy function

$$\rho(t) = \frac{c^2}{2} \log(1 + (\frac{t}{c})^2) \quad (23)$$

where c controls the convexity of the function, then we have the weight function

$$w(t) = \frac{1}{1 + (t/c)^2} \quad (24)$$

Now it seems we arrive at a typical iterative solution to the problem of robust PCA: compute the residual error with the current PCA model (14), evaluate the weight function $w(r_i^j)$ (24), compute z_i (21), eigen-decompose (22) to update the PCA model, and go back to (14) again... Obviously an iterative algorithm like this would be computationally expensive.

The reason why we have to use an iterative algorithm is that we do not know which part of a sample is likely to be outliers. However, if we have a reasonably good prototype model in hand, it is much easier to determine the outliers. In fact, the updated PCA model at each step from an incremental algorithm is good enough to serve for this purpose in most dynamic problems. Based on this idea, we can begin with estimating the parameters of the robust function, and perform the robust PCA updating in a single run.

For the robust function (23,24) we choose above, one parameter, c , needs to be determined which controls the sharpness of the robust function and hence determines the likelihood of a measurement being an outlier. In the previous studies, the parameters of a robust function are usually computed at each step in an iterative robust algorithm [8, 9] or using Median Absolute Deviation method [7]. Both of the methods are computationally expensive. Here we use a simplified method.

The first step is to estimate σ_j , the standard deviation of the j th element of the observation vectors $\{x_i\}$. Assuming that the current PCA model (including its eigenvalues and eigenvectors) is already a robust estimation from an adaptive algorithm, we approximate σ_j with

$$\sigma_j = \max_{i=1}^p \sqrt{\lambda_i} |u_{ij}| \quad (25)$$

i.e. the maximal projection of the current eigenvectors on the j th dimension (weighted by their corresponding eigenvalues). This is a reasonable approximation if we consider that PCA actually presents the distribution of the original training vectors with a hyper-ellipse in a subspace of the original space and thus the variation in the original dimensions can be approximated by the projections of the ellipse onto the original space.



Fig. 1. Comparison results of using our proposed algorithm and the conventional algorithm. From left to right are the original images, reconstruction and the weights (dark intensity for low weight) from our algorithm, reconstruction and the absolute difference images (dark intensity for large difference) from the conventional algorithm.

The next step is to express c , the parameter of (23,24), with

$$c_j = \beta \sigma_j \quad (26)$$

where β is a fixed coefficient, for example, $\beta = 2.3849$ is obtained with the 95% asymptotic efficiency on the normal distribution [10]. β can be set at a higher value for fast model updating, but at a risk of accepting outliers into the model. To our knowledge, there are no easy solutions so far to this kind of problems like determining the coefficient β .

We have compared the σ_j computed with the proposed method and that with the traditional method (the ground-truth). Experimental results indicate that the former is a good approximation of the latter. Due to page limit, these results are not included in the paper.

2.3. Algorithm

The algorithm of incremental and robust PCA is presented in Table 1. The underlying principle is very simple: we begin with an initial PCA model, compute the confidence of a new observation vector (or the likelihood to be an outlier) based on the current PCA model and weight the new vector accordingly, then perform an incremental PCA with the weighted vector. The advantages of this algorithm include:

1. Computational efficiency: it is much faster than the conventional batch-mode PCA algorithms for large scale problems, not to mention the iterative robust algorithms.
2. Model adaptability: the model can be updated online over time with new observations. This is especially important for modelling dynamic systems where the system state is variable, for example, background modelling in video surveillance which we will discuss in Section 3.
3. The computation is reasonably mild if the Cauchy function is adopted. However, even when more intensive computation like exponential and logarithm involved in the weight function w , a look-up-table (indexed by r/c) can be built for the weight item $\sqrt{w(\cdot)}$ in Equation (21) which can remarkably reduce the computation.

3. EXPERIMENTS

The proposed algorithm can be easily applied to many pattern recognition problems, especially for dynamic problems with the system status changing over time. In this section, we demonstrate the performance of the algorithm on background modelling, which is an important process for object segmentation, tracking and visual event detection.

Modelling background using PCA was firstly proposed by Oliver *et al.* [11]. By performing PCA on a sample of N images, the

Initialisation

Construct the initial PCA from the first q ($q \geq p$) observations.

Updating

FOR EACH new observation x

1. Estimate c_j from the current PCA (25,26);
2. Compute the residual error r (14);
3. Compute the weight $w(r^j)$ for each element of x (24);
4. Compute z (21);
5. Update the mean vector (2), replacing x by z ;
6. Compute y_1, y_2, \dots, y_p from the previous PCA (3);
7. Compute y_{p+1} (4), replacing x by z ;
8. Construct matrix A (5);
9. Compute matrix B (9);
10. Eigen-decompose B to obtain eigenvectors $\{v_i^{new}\}$ and eigenvalues $\{\lambda_i^{new}\}$;
11. Compute new eigenvectors $\{u_i^{new}\}$ (12).

Table 1. The algorithm of incremental and robust PCA.

background can be represented by the mean image and the first p significant eigenvectors. Once this model is constructed, one can project an input image into the p dimensional PCA space and reconstruct it from the p dimensional PCA vector. The foreground pixels can then be obtained by computing the difference between the input image and its reconstruction.

Although Oliver *et al.* claimed that this background model can be adapted over time, it is computationally intensive to perform model updating using the conventional PCA. Moreover, without a mechanism of robust analysis, the outliers or foreground objects may be absorbed into the background model. To address the two problems stated above, we extend PCA background model by applying our incremental and robust PCA algorithm. The image sequences used in the experiments are from the PET2001 datasets, a benchmark database for video surveillance. The parameters are chosen as: PCA dimension $p = 10$, size of initial training set $q = 20$, update rate $\alpha = 0.95$, and coefficient $\beta = 10$.

Figure 1 shows the comparison results on a test sequence between our algorithm (Table 1) and the conventional batch-mode PCA algorithm. It is infeasible to run the latter on the same data since they are too big to be fit in the computer memory. We randomly selected 200 frames from the sequence to perform a conventional batch-mode PCA. Then the trained PCA was used as a fixed background model.

It is noted that our algorithm successfully captured the background changes while the fixed PCA model failed with noticeably ghost effect in the reconstruction and the false foreground detection. In this experiment, our algorithm achieved a frame rate of 5 fps on a 1.5GHz Pentium IV computer (with JPEG image decoding and image displaying).

To illustrate the importance of robust analysis, we show the

first three eigenvectors of the PCA background models of using and not using robust analysis in Figure 2. It is noted that the non-robust algorithm unfortunately captured the variation of outliers, most noticeably the trace of pedestrians and cars on the walkway appearing in the images of the eigenvectors. This is exactly the limitation of conventional least-squares based PCA as the outliers usually contribute more to the overall mean squared error and thus deviate the results from desired. On the other hand, the robust algorithm correctly captured the variation of the background.

It can be further illustrated in Figure 3 which shows the values of the first dimension of the PCA vectors computed with the two algorithms. It is observed that the non-robust algorithm presents a fluctuant result because it struggled to compensate the large error from outliers by severely adjusting the values of model component, especially when significant activities happened during frames 1000-1500, while the robust algorithm achieves a steady performance.

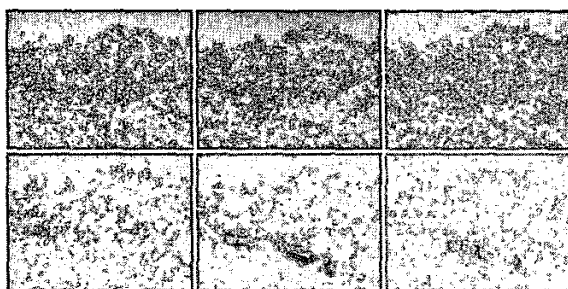


Fig. 2. The first three eigenvectors obtained from the robust algorithm (upper row) and non-robust algorithm (lower row). The intensity values have been normalised to $[0, 255]$ for illustration purpose.

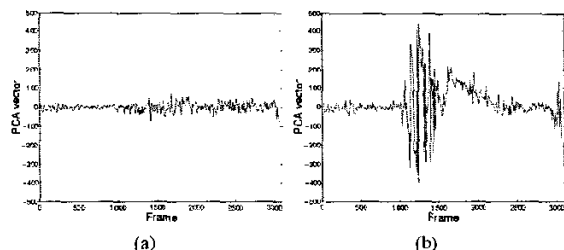


Fig. 3. The first dimension of the PCA vector computed on the same sequence in Figure 1 with robust analysis (a) and without (b).

4. CONCLUSIONS

Learning PCA incrementally and robustly is an interesting issue in pattern recognition and image processing. However, this problem has only been addressed separately in the previous studies, i.e. either incrementally or robustly. In this work, we have developed an incremental PCA algorithm, and extended it with efficient robust analysis.

As we do not know which part of a sample is outliers, traditional M-estimation based robust algorithms usually perform in an iterative manner which is computational expensive. However, if a PCA model is updated incrementally, the current model is usually sufficient to be used for outlier detection for new observation vectors. This is the underlying principle of our proposed algorithm.

Estimating the parameters of a robust function is usually a tricky problem for robust M-estimation algorithms. In this work, we have solved this problem by approximating the parameters directly from the current eigenvectors/eigenvalues, so that the computation of the robust algorithm is only slightly more than an incremental algorithm.

As an example, we demonstrate the performance of the algorithm on dynamic background modelling, with a comparison to the conventional batch-mode PCA algorithm and the non-robust algorithm. Improved results have been achieved in the experiments. This algorithm can be readily applied to many large scale PCA problems, especially dynamic problems with system status changing over time.

5. REFERENCES

- [1] P. Gill, G. Golub, W. Murray, and M. Saunders, "Methods for modifying matrix factorizations," *Mathematics of Computation*, vol. 28, no. 26, pp. 505-535, 1974.
- [2] J. Bunch and C. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, no. 2, pp. 131-152, 1978.
- [3] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkler, and H. Zhang, "An Eigenspace update algorithm for image analysis," *Graphical Models and Image Processing*, vol. 59, no. 5, pp. 321-332, 1997.
- [4] P. M. Hall, A. D. Marshall, and R. R. Martin, "Incremental eigenanalysis for classification," in *British Machine Vision Conference*, P. H. Lewis and M. S. Nixon, Eds., 1998, pp. 286-295.
- [5] Lei Xu and A. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 131-143, 1995.
- [6] K. Gabriel and C. Odoroff, "Resistant lower rank approximation of matrices," in *Proceedings of the Fifteenth Symposium on the Interface*, J. Gentle, Ed., Amsterdam, Netherlands, 1983, pp. 304-308.
- [7] F. De la Torre and M. Black, "Robust principal component analysis for computer vision," in *IEEE International Conference on Computer Vision*, Vancouver, Canada, 2001, vol. 1, pp. 362-369.
- [8] Peter J. Huber, *Robust Statistics*, John Wiley & Sons Inc, 1981.
- [9] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust Statistics*, John Wiley & Sons Inc, 1986.
- [10] Zhengyou Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59-76, 1997.
- [11] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831-841, 2000.