Regressive Principal Component Analysis

Vladimir Bochko
University of Vaasa, Yliopistonkatu 10, P.O Box 700, FI-65101, Finland
Tel: + 358 44 3364 688, vbochko@uva.fi

KEY WORDS: principal component analysis, regression, nonlinear data structures

ABSTRACT

In this paper, we introduce Regressive Principal Component Analysis (RPCA). RPCA combines two techniques: Principal Component Analysis (PCA) and regression which easy to implement. RPCA has many features like PCA and capable to capture the nonlinear data structure. The method is efficient to work with a data with a low degree of nonlinearity in automation control and computer graphics.

1 INTRODUCTION

The dimensionality reduction problem solves machine learning problems and helps to obtain accurate data analysis in regression and classification. The dimensionality reduction is used for visualizing structure in data, denoising and extracting meaningful features.

For feature extraction to represent a high-dimensional data in a low dimensional space principal component analysis (PCA) and nonlinear manifold learning techniques are utilized. PCA is an efficient technique when the analysed variables have a Gaussian distribution and data structure is linear /3/. PCA is also capable to capture the nonlinear data structure and reduce data dimension. However, the number of components in this case should be larger than intrinsic dimensionality of data. This means that PCA non-optimally represents the nonlinear data. The manifold learning technique is used for reducing dimensionality in the case of nonlinear data structures. The manifold learning technique includes Locally-Linear Embedding (LLE), ISOMAP, autoassociative neural network, non-linear PCA, kernel PCA and others /2, 5, 7, 10, 11/. The manifold learning techniques have drawbacks. The techniques may not be robust in computing, may not converge, computationally demanding, may not work even with moderate scale data and work only with well-defined data structures because of sensitivity to noise. All these problems make the use of manifold learning techniques difficult. There is clearly a niche between these linear and nonlinear approaches. The technique keeping advantages of PCA and working with nonlinear data is desirable.

Many image sets and videos (patient breath images, bird flying in the direction of a camera, and images of rotated object acquired by camera from different viewpoints) originally presented in high-dimensional space have a few degrees of freedom /9/. In automation, many mechanisms have a few degrees of freedom as well. Thus, the low dimensional subspaces can be useful in applications.

In this paper, we introduce Regressive Principal Component Analysis (RPCA). RPCA has many features like PCA and capable to capture the nonlinear data structure. The method limitation is that it works with a data with a low degree of nonlinearity. In automation control, the nonlinearity related to friction effect or the value limiting devices are usually low degree. We study mostly low dimensional subspaces. We give analytical and neural

network implementation of RPCA. This study shows the wide range of the possible RPCA use with a different nonlinear data including the color data introduced earlier /4/. Our experiments are conducted with synthetic data, and data used in computer graphics and automation control.

2 METHODOLOGY

The proposed approach RPCA is based on concept of nested subspaces and uses two techniques: PCA and regression. First, we use PCA to reduce data dimensionality to intermediate-dimension (first) subspace, i.e. lower dimension subspace preserving the data nonlinearity. Then, we find a mapping between the leading principal components (PC) to weak principal components related the subspace. The leading PCs are the second nested subspace. Finally, to represent a nonlinear data structure we retain only the leading PCs and mapping parameters. The method limitation is that it may work only with a low degree of nonlinearity when data, i.e. PCs exhibit one-to-one mapping. The difference between RPCA and manifold learning technique is that the leading PCs of RPCA are poor represent the intrinsic data dimension. However, in applications the RPCA efficiency may be close to manifold learning. For example in data compression, we may have one nonlinear component of manifold learning versus one equal size leading principal component of RPCA and mapping parameters, that is a rather small amount of data. RPCA has not computation problems inherent to machine learning like non-robust computing, long computation time and sensitivity to noise. In addition, RPCA is capable to make not only forward data mapping but also the inverse mapping, that many manifold learning methods cannot provide.

2.1 Generic RPCA

Assume that a centered data is presented by a matrix X with a size $n \times t$, where n is a number of variables and t is a number of observations. We sequentially reduce the dimension calculating the first subspace, which dimension is $d \leq n$ and, then, the second subspace with dimension $p < d$. For the first reduction, we use PCA based on singular-value decomposition (SVD) as follows:

$$X = USV^T,$$

where the columns of U contain the eigenvectors of XX^T. We will use only the first d eigenvectors U_d corresponding to the largest d eigenvalues (singular values). Thus, data encoding is as follows:

$$Y_d = U_d^T X,$$

where Y_d is a matrix with a size $d \times t$. Then, we select p component among d component of Y_d corresponding to the largest eigenvalues. The selected p components represent the second subspace and utilized for approximation the rest non-selected components of Y_d. The matrix Y_d is comprised of two matrices: Y_p (the selected components) and the matrix Y_{d-p} (the rest components). The approximation of Y_{d-p} by Y_p is obtained using regression for which the regression function $f()$ and the mapping parameters B are defined. There are many regression techniques suitable in our case: Ordinary regression, bridge regression, multilayer perceptron regression, Gaussian process regression and radial basis function regression /2, 8/. In summary for data reconstruction, we calculate the mean vector and the first d eigenvectors, select the regression function and define its parameters.

To reconstruct data, the regression is first used

$$\hat{Y}_{d-p} = f(Y_p, B).$$

Then, we compose matrices $\hat{Y}_d = [Y_p, \hat{Y}_{d-p}]^T$, and reconstruct the data

$$\hat{X} = U_d \hat{Y}_d.$$

2.2 RPCA using multiple linear regression

Multiple Linear Regression (MLR) /2/ is one of the simple and efficient way to use with RPCA. In this case

$$Y_{d-p}^T = Y_p^T B + E,$$

where Y_{d-p}^T is a matrix which columns are output components or responses, Y_p^T is a matrix which columns are input components used for approximation, B is a matrix of regression coefficients the columns of which are regression vectors and E is an error.

The solution, i.e. the matrix B, is calculated as follows;

$$B = (Y_p Y_p^T) Y_p Y_{d-p}^T.$$

Finally, the mapping is as follows:

$$\hat{Y}_{d-p}^T = Y_p^T B.$$

If we want to approximate nonlinear data then we need to use a polynomial fit extending Y_p^T. Let us consider an example related to computer graphics. The dichromatic reflection model is widely used to describe the color image regions with variable intensity but approximately constant hue /4/. The model represents the global nonlinear data using a piece-wise linear data structure. The two linear clusters (or data structures) are specular reflection and body reflection. The model suggests that the data in RGB space is span by two leading eigenvectors, i.e. the first subspace called the dichromatic plane, and two linear clusters locate in the dichromatic plane. For these two clusters RPCA uses the data projection on the first eigenvector to approximate the projection on the second eigenvector. RPCA replaces the piece-wise linear model with a nonlinear model /1/. In this case we have n = 3, d=2, and p =1.

Thus, the data encoding is $Y_d^T = [y_p^T, y_{d-p}^T]$, where columns of Y_d^T y_p^T and y_{d-p}^T are data projections. The relationship between y_p^T and y_{d-p}^T is nonlinear. To approximate y_{d-p}^T using y_p^T we exploit regression. To capture nonlinearity, we use a nonlinear fit, i.e. polynomial extension $Y_p^T = [1, (y_p^T)^1, (y_p^T)^2, \dots, (y_p^T)^k]$. Now we have to select the order of polynomial. The lower order may produce poor approximation while the higher order may lead to overfitting. The order 3-5 is suitable in this case. Thus, finally extended Y_p^T is

$$Y_p^T = \left[1, (y_p^T)^1, (y_p^T)^2, (y_p^T)^3\right],$$

where 1 is a unit entry vector to provide a bias.

2.3 Neural network RPCA

Here we show a possible RPCA implementation using an autoassociative neural network (ANN). We utilize the multilayer perceptron (backpropagation algorithm). The network training include mapping the input data onto the same output data providing the minimum of sum-of-squares error (Fig. 1).

In similar example with the conventional autoassociative neural network the data is reduced to a nonlinear component /2/. In this case, to make this component nonlinear we have to introduce the addition layer. The additional layer locates between the input layer and the first layer and consists of two nonlinear neurons. The proposed autoassociative neural network is simply because it has not the layer with nonlinear neurons and the component (first neuron) is still linear (like PCA) (Figure 1). The two hidden (nonlinear) neurons capture the nonlinearity when data is reconstructed.

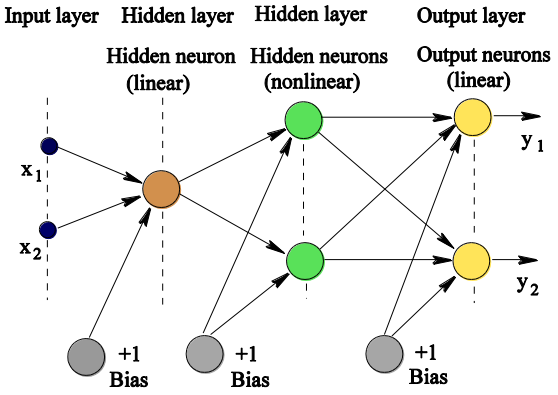


Figure 1. Autoassociative neural network RPCA. The input 2D data is mapped onto the low-dimension space described by one linear component and, then, is reconstructed using two nonlinear neurons and two output linear neurons.

Depending on data, we design the neural network determining the number of layers and the number of neurons in each layer. The given neural network stricture is for nonlinear 2D data (Figure 1).

3 EXPERIMENTS

We conducted several experiments with synthetic and real data. In all experiments, the polynomial of the third order was used to provide nonlinear fit. The first experiment was conducted with a neural network and toy data. The second and third experiments were conducted with the Helix and Hemisphere. In the fourth experiment, the image colorization algorithm was used. Finally, RPCA was tested with a control system.

The first experiment relates to the autoassociative neural network described in Section 2.3 and utilized for the toy data (Figure 2). We use the uniformly distributed data $x_1$in the range [0.0, 1.5] and $x_2 = x_1^3$. The Gaussian noise N (0.0, 0.07^2) was added to both components. Then, the data was rotated counter clockwise around the point (0.7, 0.7) by an angle 15 degrees. In this case, $n = d = 2$ and $p = 1$.

The reconstructed data correctly show the intrinsic dimension (Figure 2). This result is impossible to achieve using only regression because the mapping between variables is one-to-many. However, the nonlinearity is correctly reproduced using RPCA because the mapping between the first and second principal components is one-to-one.

In the next experiment, we generated the Helix as follows. The source x_1 has 50 evenly distributed points ranging from 0 to 2π. Then the z-axis is equated with the source, the x-axis is $x_2 = \sin x_1$, and the y-axis is $x_3 = \cos x_1$(Figure 3a, solid line). The Gaussian noise N (0.0,0.05^2) was added to all components. In this case, $n = d = 3$ and $p = 1$. The RPCA data reconstruction is rather good (Figure 3a, dots).

In the third experiment, we designed the Hemisphere with a unit radius and center at the axis origin. We generated 32 evenly distributed points for θ in the range $[-\pi, \pi]$ and 16 evenly distributed points for φ in the

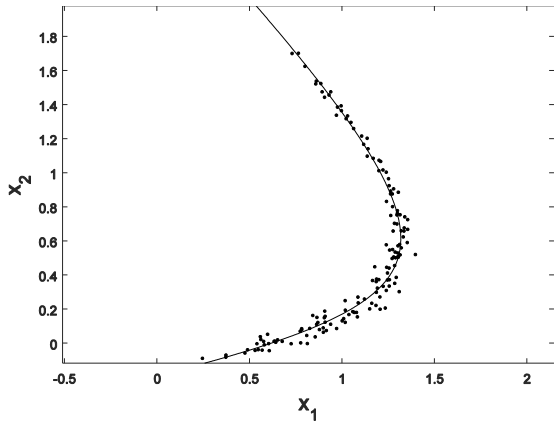
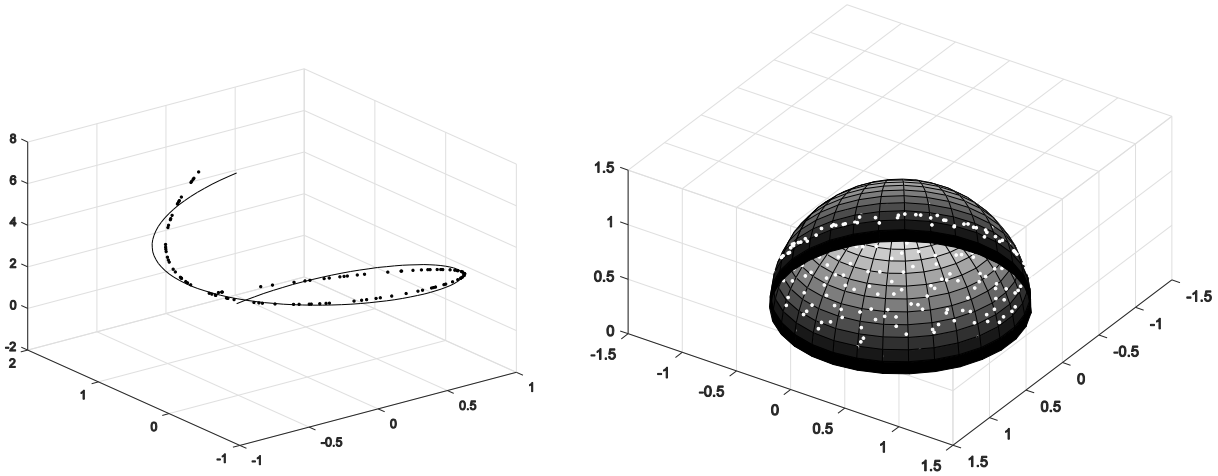


Figure 2. Autoassociative neural network RPCA. The calculated subspace is shown by a solid curve. The original data is shown by dots.

range $[0, \pi/2]$. Then the x, y and z-axis are as follows: $x = \cos\varphi \cos\theta$, $y = \cos\varphi \sin\theta$, and $z = \sin\varphi$. For all trhee components the Gaussian noise was added N $(0.0, 0.03^2)$ In this case, $n = d = 3$ and $p = 2$. The data was successfully reconstructed using RPCA (Figure 3b).



a) b)

Figure 3. Reconstruction. a) Helix. The underlying subspace is shown by a solid line. The reconstructed subspace is shown by dots. b) Hemisphere. The underlying subspace is shown by a solid surface. The reconstructed subspace is shown by dots. The dots locate nearby with internal and external sides of the hemisphere surface.

The fourth experiment relates to computer graphics using gray-level image colorization. The dichromatic reflection model was used /4/. In this case we have the following dimensionality: $n = 3$, $d = 2$ and $p = 1$. The

gray-level image Apple has a size 300×200 pixels (Figure 4a). The colorization based on RPCA (Figure 4c) is superior to PCA (Figure 4b). The saturated color and highlight are visually better reproduced by RPCA.

Finally, we conducted experiment with a task related to automation control where the controller is subject to saturation (Figure 5a) /6/. The equivalent state model for the plant is as follows:

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -4x_2 + 2u,$$

where x_1 and x_2 are state variables: output/position and velocity, respectively, u is a control signal.

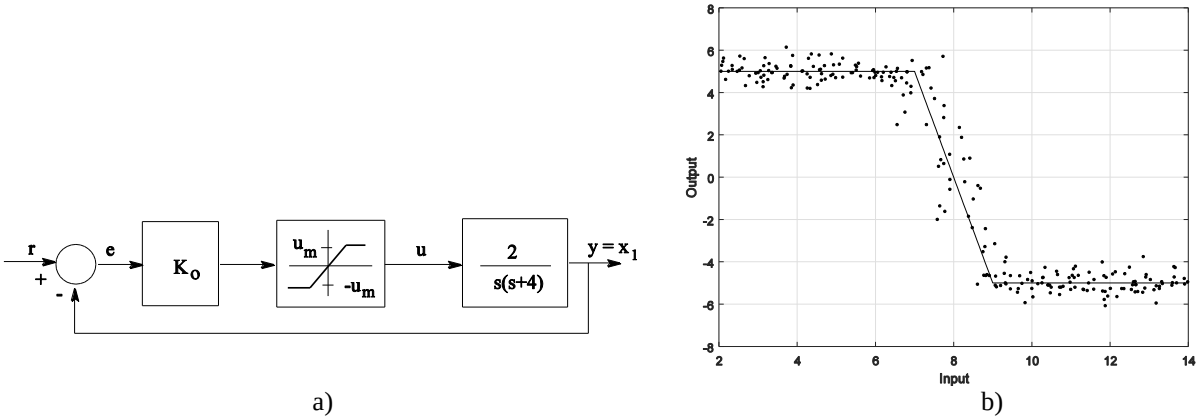
a) b) c)

Figure 4. Image colorization. a) Original gray-level image. b) Colorization using PCA. c) Colorization using RPCA. RPCA correctly captures the data nonlinearity and better reproduces saturated color and highlight than PCA.

The controller is modeled using a piece-wise liner characteristic:

$$u = +u_m \quad \text{if} \quad K_o(r - x_1) \geq u_m,$$
$$u = +u_m \quad \text{if} \quad K_o(r - x_1) \leq -u_m,$$
$$\text{or} \quad u = K_o(r - x_1) \quad \text{if} \quad K_o|r - x_1| < u_m.$$

We use the reference $r(t) = 8u(t)$, $K_o = 5$ and $u_m = 5$, where $u(t)$ is a unit step function.

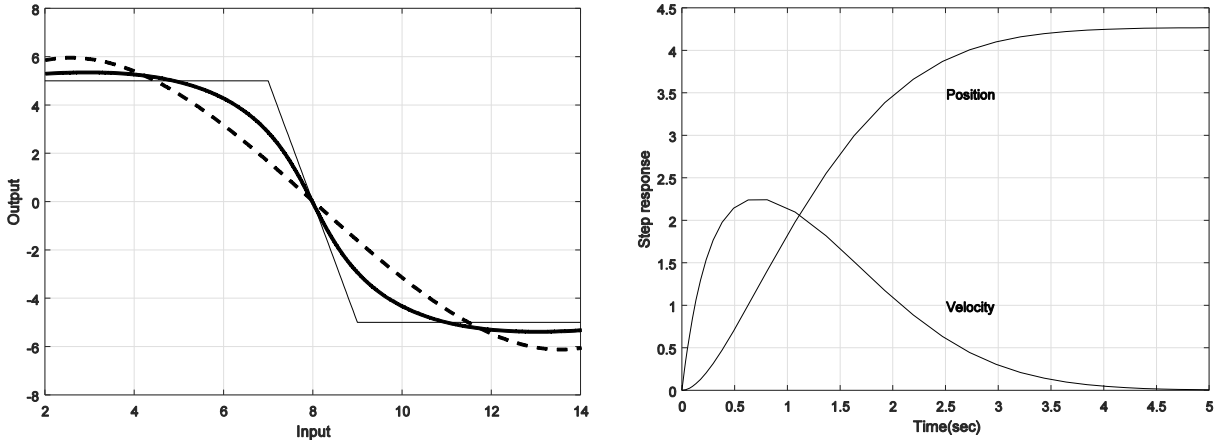


a) b)

Figure 5. Control system. a) System with limiting. b) Modeling the saturation characteristic.

One way of solution could be to use three solutions related to linear regions. However, it is better to use one single solution based on global nonlinear characteristic. In addition, the smoothed characteristic close to reality is desirable /6/. To imitate measurements errors, we added Gaussian noise N $(0.0,0.4^2)$ to the original piece-wise characteristic (Figure 5b). RPCA provides denoising and obtaining the smooth curve. The regression method

solves the same task using the nonlinear fit with a third order polynomial. In this case, the result is not so accurate in comparison with RPCA (Figure 6a). To solve the task with RPCA approximation we used the Matlab **ode23** (Runga-Kutta) algorithm (Figure 6b).



a) b)

Figure 6. Control system. a) RPCA approximation (thick solid line), underlying piece-wise linear characteristic (thin solid line) and regression based approximation (dash line). b) Step response for velocity and position.

4 CONCLUSIONS

We developed and completed the method RPCA for the data dimensionality reduction. RPCA is simple for implementation and based on PCA and linear regression. The proposed method can be easily modified to obtain desirable features utilizing different kinds of PCA (PCA, probabilistic PCA, Bayesian PCA and others) and regression (ordinary regression, bridge regression, kernel regression and others). The method find the efficient solution for the data with a low degree of nonlinearity. In our future work, we will modify the algorithm for the higher degree of data nonlinearity.

5 REFERENCES

1. Bochko V., Välisuo P., Alho T. M. R., Sutinen S., Parkkinen J. and Alander J., Medical Image Colorization Using Learning, The Fifth European Conference on Color in Graphics, Imaging and Vision CGIV 2010, Joensuu, Finland, pp. 70-74, 2010, ISBN: 978-0-89208-291-9.

2. Bishop C. M., Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.

3. Jolliffe, I. T., Principal Component Analysis, 2nd edition, Springer, 2002.

4. Klinker G. J., Shafer S. A., and Kanade T., A Physical Approach to Color Image Understanding, International Journal of Computer Vision, Vol. 4, No. 1, pp. 7-38, 1990.

5. Lappalainen H. and Honkela A., Bayesian Nonlinear Independent Component Analysis by Multi-Layer Perceptrons, in Advances in Independent Component Analysis, M. Girolami, Editor, Springer-Verlag, pp. 93-121, 2000.

6. Lewis P. H., C. Yang C., Basic Control Systems Engineering, Prentice Hall, 1997.

7. Mika S., Schölkopf B., Smola A., Müller K.-R., Scholz M., and Rätsch G., Kernel PCA and De-noising in Feature Spaces, In: Advances in Neural Information Processing Systems 11, ed. by M. S. Kearns and S. A. Solla and D. A. Cohn, Cambridge, MA, MIT Press, pp. 536 – 542, 1999.

8. Nabney I. T., Netlab Algorithms for Pattern Recognition, Springer, 2002.

9. Pless R., Souvenir S., A Survey of Manifold Learning for Images, IPSJ Transactions on Computer Vision and Applications, vol. 1, pp. 83-94, 2009.

10. Roweis S. T. and Saul L. K., Nonlinear Dimensionality Reduction by Locally Linear Embedding, Vol. 290, pp. 2323-2326, 2000.

11. Tenenbaum J. B., de Silva V. and Langford J. C., A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science, Vol. 290, pp. 2319-2323, 2000.