NEUROEVOLUTIONARY PSEUDO-PRINCIPAL COMPONENTS ANALYSIS[1]

Y. Tsoy

Tomsk Polytechnic University, 84, Sovetskaya Str., Tomsk, 634034,
Tomsk State University of Control Systems and Radioelectronics, 40, Lenina Ave., Tomsk, 634050
yurytsoy@gmail.com

This paper concerns application of neuroevolutionary approach for dimensionality reduction, which leads to the results analogous to that of PCA except that eigenvectors are calculated approximately. Thus the presented algorithm is called pseudo-PCA. Experimental results for several classification problems of different dimensionalities show that such an approximation doesn't tend to have much influence on the classification accuracy. Since pPCA is an evolutionary algorithm it can be effectively parallelized, which is promising from the speed of computations perspective.

Introduction

Principal Components Analysis (PCA) is one of the most popular methods for dimensionality reduction for pattern recognition problems [1]. It exploits simple and yet powerful idea that certain linear subspaces of the features space do not contribute much in the data and thus can be thrown away without significant damage to the initial information. PCA can also be considered as a sort of lossy compression algorithm, because linear subspaces to be thrown away contain some bit of information.

Technically PCA requires finding of eigenvalues and eigenvectors of autocorrelation matrix **C** for training data. This implies usage of numeric methods and is significantly eased by the fact, that **C** is symmetric, so that special more effective numeric methods can be applied [2]. One of the biggest issues here is that such methods are hardly parallelizable due to iterative nature.

At the same time a hypothesis can be stated that it's not that important to compute exactly principal eigenvectors, but to find low-contributing ones associated with small eigenvalues. In this paper a new neuroevolutionary approach for removal of linear subspaces containing small portions of information is presented. It differs from known neural algorithms for computation of PCA, which are based upon Hebbian learning idea [3], since it doesn't exploit correlation between neural nodes' outputs. The method presented is not aimed at exact calculation of eigenvectors hence it will be referred as pseudo-PCA (pPCA).

Idea of the Method

As it was stated above the presented method removes low-informative linear subspaces. First, note that PCA can be reformulated to find such linear subspaces, for which orthogonal projection of data points has maximal variance. In other words, linear subspaces with small variance of data points projections do not fit PCA. The question is whether we can throw away such subspaces for sure without harming the consequent steps? The answer is positive due to the following

Proposition 1: Let $\mathbf{X}_i = \{\mathbf{X}_i, i = 1,..., N\}$, $\mathbf{X}_i \in R^n$ be a set of data points and $\mathbf{Q}_i = \{\mathbf{q}_i, i = 1,..., n\}$ is an orthogonal basis in R^n. Denote $proj_{\mathbf{q}_i}(\mathbf{X})$ as projection of data points from $\mathbf{X}$ onto coordinate vector $\mathbf{q}_i$, and

[1] The research is supported by the Russian Foundation for Basic Researches grants (projects no. 09-08-00309-a, 11-07-00027-a).

$Var(proj_{\mathbf{q}_i}(\mathbf{X}))$ as a variance of correspondent projections. Then summation over all dimensions

$$\sum_{i=1,\dots,n} Var(proj_{\mathbf{q}_i}(\mathbf{X})), \quad (1)$$

is constant and doesn't depend on **Q**.

Sketch of the proof. The proposition can be proved by direct substitution of expression for variance into the sum and after some algebra it can be shown that resulting expression doesn't contain **Q**.

This proposition says, that since sum of projection variances is constant, then if columns $\hat{\mathbf{Q}}$ are estimates for eigenvectors, the low-contributing columns in $\hat{\mathbf{Q}}$ will be even less significant when coordinates of "primary" eigenvectors' estimates are defined more precisely. Thus we can throw away basis vectors, which do not fit some criterion, as non-informative without much harm for elaboration of coordinates of more significant vectors. This has certain benefits from the computational point of view because dimensionality of the problem is reduced when low-contributing basis vectors are removed.

In this paper the criterion for removal of low-informative subspaces is (eigenvectors are to be sorted by decrease of variance):

$$\frac{Var(proj_{\hat{\mathbf{q}}_0}(\mathbf{X}))}{Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X}))} < \tau, \quad (2)$$

where $\hat{\mathbf{q}}_i$ - estimate of the *i*-th eigenvector, τ - is a threshold. Typical values for τ are 5, 10, 20, …. Note that under this approach it is possible to truncate low-informative subspaces without knowing exact coordinates of principal eigenvectors. That's why the presented method is called pseudo-PCA: it removes possibly unimportant linear subspaces (with some error), but doesn't compute principal components.

The above argumentation gives rise to the following neuroevolutionary algorithm:

1. **Initialize** random population, each individual is a candidate solution for pPCA.
2. **Evaluate** each individual using the following fitness function:
   $f = \sum_i Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X})) \to \max$
   a. Compute mean variances $\overline{var}$ of projections onto eigenvectors' estimates.
   b. Starting from the last element in $\overline{var}$ check whether criterion (2) is satisfied. If it does for some *k*-th element, then remove correspondent genes from all individuals in population. If not, then proceed to **Step 3**.
3. **Selection**.
4. **Crossing** and **Mutation**.
5. If algorithm run's is completed then proceed to **Step 6**, otherwise proceed to **Step 2**.
6. **Return** the best found individual.

Special operators are used for evaluation of fitness and crossing.

Fitness evaluation procedure

Fitness evaluation procedure can be expanded in the following way (for the *i*-th individual):

1. Assign genes of individual to Artificial neural network (ANN) with linear nodes.
2. Apply Gram-Schmidt orthogonalization to ANN weights.
3. Compute responses of ANN for each training sample.
4. Compute variances of ANN outputs.
5. Sort ANN nodes by the decrease of variances.
6. Copy obtained vector of weights back into chromosome.

Sorting of ANN outputs by responses' variance is required to order basis vectors, correspondent to the nodes' weights, by their contribution to training data representation. Note that evaluation of each individual is independent and can be performed in parallel.

Crossing

The algorithm also uses a special crossover operator, which employs an idea from the Euler's approximation.

Let *i*-th and *j*-th individuals are crossed each representing combination of all weights of ANN ($\mathbf{w}_i$ and $\mathbf{w}_j$ respectively). Suppose that the *i*-th individual is the better one. Each vector of weights is split into N_O parts, where N_O – is a number of ANN outputs:

[1] The research is supported by the Russian Foundation for Basic Researches grants (projects no. 09-08-00309-a, 11-07-00027-a).

$$\mathbf{w}_i = \{\mathbf{w}_i^{(k)}, k = 1,..., N_O\},$$

so that each *k*-th part corresponds to weights of the *k*-th output node. With each part $\mathbf{w}_i^{(k)}$ a weight $v_i^{(k)}$ is associated, which equals to the variance of projection of training data samples onto the correspondent coordinate vector.

The crossing of two individuals is performed part-wise to produce one offspring using:

$$\mathbf{c}^{(k)} = \mathbf{w}_i^{(k)} + \frac{|\tilde{v}_i^{(k)} - \tilde{v}_j^{(k)}|}{\left\|\mathbf{w}_i^{(k)} - \mathbf{w}_j^{(k)}\right\|}(\mathbf{w}_i^{(k)} - \mathbf{w}_j^{(k)}), \quad (3)$$

where $\mathbf{c}^{(k)}$ – *k*-th part of the offspring's chromosome, $\tilde{v}_i^{(k)} = v_i^{(k)}/(v_i^{(k)} + v_j^{(k)})$ and $\tilde{v}_j^{(k)} = v_j^{(k)}/(v_i^{(k)} + v_j^{(k)})$ – normalized weights of parent parts, $\|\mathbf{x}\|$ – Euclidian norm of **x**.

The fraction in (3) is used to approximate the absolute value of gradient of *k*-th part weight so that overall expression (3) can be treated as linear approximation for the update of *k*-th part moving from the point $\mathbf{w}_i^{(k)}$.

Experiments Description

Main goal of the experimental study is twofold:

1. It is important to find out whether efficient dimensionality reduction is possible.
2. Since pPCA doesn't yield linear subspaces associated with the principal components it's also important to know how this affects classification accuracy.

Testing of the proposed pPCA method will be conducted using several classification problems from the Proben1 set [4], namely (dimensionality of the features space is given in brackets): *cancer1* (9), *card1* (51), *diabetes1* (8), *glass1* (9), *heart1* (35), *horse1* (58), *soybean1* (82), *thyroid1* (21).

NE approach will be used for reduction of the features space dimensionality and after that transformed objects descriptions will be used to train feed-forward ANN using traditional gradient training.

The following values for criterion τ for removal of non-informative linear subspaces according to (2) to be considered: 5, 10, 20.

During each run a pPCA solution to be found and for this solution 10 ANNs will be trained, from which only one is selected using classification error on a validation data set. This winning ANN is used for classification of samples from a test data set. For each problem 10 runs are performed and mean classification accuracy is used for comparison and analysis.

EA run duration to find a particular pPCA solution is 50 generations and each ANN for classification is trained using RPROP algorithm for 100 epochs.

Results of Experiments and Discussion

Classification error values on a test set are given in the Table 1. Also a number of ANN outputs in result of neuroevolutionary pPCA is given in brackets to evaluate dimensionality reduction.

Table 1. Classification errors (%) for different values of τ.

Problem	$\tau = 5$	$\tau = 10$	$\tau = 20$
cancer1 (9)	2.59 (1)	2.53 (1.3)	1.78 (6.3)
card1 (51)	14.77 (24.1)	16.22 (46.4)	15.99 (50.7)
diabetes1 (8)	26.56 (7.6)	26.67 (8)	26.51 (8)
glass1 (9)	40.38 (5.1)	35.09 (7)	31.89 (8.2)
heart1 (35)	21.13 (23.3)	20.00 (32.4)	20.00 (35)
horse1 (58)	29.45 (36.7)	30.44 (57.5)	32.97 (58)
soybean1 (82)	23.00 (11)	10.71 (32.7)	8.53 (66.1)
thyroid1 (21)	7.31 (5.2)	6.88 (10.6)	6.26 (18)

For comparison Table 2 contains the best classification errors from [4-6] for the same problems. Results from [4] are obtained for traditional approach for ANN training, from [5] – using genetic algorithm only, and from [6] – using pruning methods for neural networks.

Table 1. Classification errors (%) for other some approaches on the Proben1 set.

Problem	The best classification errors, %		
	[4]	[5]	[6]
cancer1	1.38	1.24	1.1
card1	14.05	14.27	13.7
diabetes1	24.10	23.70	20.8
glass1	32.7	47.62	30.2
heart1	19.72	21.87	18.5
horse1	29.19	26.44	26.9

[1] The research is supported by the Russian Foundation for Basic Researches grants (projects no. 09-08-00309-a, 11-07-00027-a).

soybean1	9.06	8.47	N/A
thyroid1	2.32	6.12	5.7

Comparing the results it can be seen that for most problems under consideration reduction of dimensionality due to pPCA provides comparable classification accuracy with hand-tuned ANNs trained in a "traditional" way. At the same time a reduction of dimensionality was quite significant (see cancer1 or soybean1 results). Figures 1 and 2 show examples of change of averaged mean dimensionality and variances of projections onto the eigenvectors estimates with respect to time.

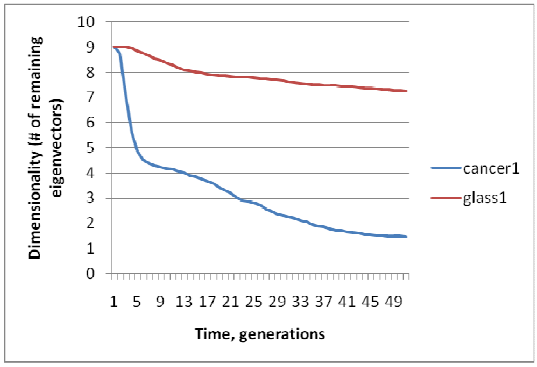

Fig. 1. Change of averaged dimensionality for *cancer1* and *glass1* problems. $\tau = 10$.

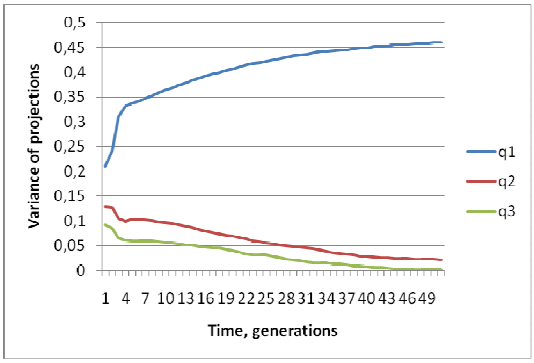

Fig. 2. Change of averaged variances of projections of data points onto the first 3 eigenvectors estimates for cancer1 problem. $\tau = 10$.

Relying upon the obtained results a very interesting conclusion can be made:

> *There are cases when it's not necessary to know exactly principal components of autocorrelation matrix to perform a reliable dimensionality reduction.*

Interesting question is what the price for use of approximate principal components is. If exact non-informative linear subspaces were ripped off, then transformed features would be living in a "correct" subspace and hence there is a linear transformation, which converts remaining eigenvectors estimates obtained by pPCA to the exact solutions for PCA. But if dimensionality reduction was made for inexact eigenvectors then there's a transformation error for pPCA comparing with the PCA. The measurement of this error is somewhat tricky and is an open question, which should be studied thoroughly in order to understand features of the pPCA.

Conclusion

The paper presents novel way for dimensionality reduction using pseudo-PCA. The method is based upon application of neuroevolutionary approach for feed-forward linear neural networks without hidden nodes and uses special procedure for fitness estimation and a crossover operator.

Experimental results show that even though the reduction of features space dimensionality was performed along approximate eigenvectors the obtained classification results are comparable with that of ANNs trained in more traditional ways.

Interesting questions for the future research are parallelization of pPCA algorithm and analysis of the error in computation of eigenvectors.

References

1. Gorban A., Kegl B., Wunsch D., Zinovyev A. (Eds.) Principal Manifolds for Data Visualisation and Dimension Reduction. - LNCSE 58. - Berlin, Heidelberg, New York: Springer, 2007.
2. Golub G.H., Van Loan C.F. Matrix Computations. Second Edition. - Baltimor, London: The John Hopkins University Press, 1989.
3. Oja E. Simplified neuron model as a principal component analyzer // Journal of Mathematical Biology, 1982, vol. 15, no. 3, pp. 267-273.
4. Prechelt L. PROBEN1 - a set of neural network benchmark problems and benchmarking rules. Technical Report 21/94. - Universitat Karlsruhe, Karlsruhe, Germany, 1994.
5. Barrera F.Y. Busqueda de la estructura optima de redes neurales con Algoritmos Geneticos y Simulated Annealing. Verificacion con el benchmark PROBEN1 // Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 2007, Vol. 11, No. 34, pp. 41-61.
6. Weman N. Empirical Investigation of the Effect of Pruning Artifical Neural Networks With Respect to Increased Generalization Ability. - Linkopings universitet (Sweden), 2010.

[1] The research is supported by the Russian Foundation for Basic Researches grants (projects no. 09-08-00309-a, 11-07-00027-a).