Proc. of 1992 IEEE-INNS International Joint Conference on Neural Networks (IJCNN92), June 7-11, 1992, Baltimore, MA, Vol. I, pp.812-817.

Robust PCA Learning Rules Based on Statistical Physics Approach[1]

Lei Xu[1,2] and Alan Yuille[1]
1. Harvard Robotics Laboratory, Harvard University, USA
2. Dept. of Mathematics, Peking University, P.R.China
The correspondence address is given at the bottom of this page

Abstract This paper adapts statistical physics approach to the problem of robust *Principal Component Analysis (PCA)*. First, some common used PCA learning rules are connected to some energy function which is further generalized by adding a binary decision field with a given prior distribution so that outliers are considered. Second, the generalized energy is used to define a Gibbs distribution and to derive an effective energy function which is further used to derive learning rule for robust PCA. The experimental results have shown that our robust rules have improved the performances of the existing PCA algorithms significantly.

1 Introduction

Principal Component Analysis (PCA) is an essential technique for data compression and feature extraction, and has been widely used in statistical data analysis, communication theory, pattern recognition and image processing. Oja (1982) found that a simple linear neuron with a constrained Hebbian learning rule can extract the *Principal Component (PC)* from stationary input data [1], which built up the first connection between self-organizing rule in neural networks and PCA technique. Recently, there has been increasing interest in the study of connections between PCA and neural network. A symmetrical error correcting learning rule is proposed by William (1985) for a two layer net which can discover the subspace spanned by the first k PCs. Multi-Layer Perceptron (MLP) in *supervised autoassociative* mode have been suggested for data compression (Cottrell86) and have been shown to be closely connected to PCA (Bourlard88, Baldi89). A number of unsupervised rules for extracting PCs or their spanned subspace have been also proposed and studied (Kammmen & Yuille88, ,Rubner89, Sanger89, Hryycej90, Linsker90, Oja91a, Xu91a, Biald91). The relationship between PCA and the emergence of feature analyzing properties in cortex field of biological systems have been revealed and further studied (Linsker86, Barrow87, Mackey87, Yuille89a). Furthermore, some extensions of PCA have been also made (Kung90, Xu91b,c&d. Oja91b).

However, the performances of these existing algorithms will deteriorate significantly under the presence of outliers. The problem is essentially important to real applications since data usually contains outliers in practice. Presently, little attention has been paid to this problem in neural network literature, perhaps partly due to the hardness incurred when outliers are taken into consideration.

Recently, there are a number of successes in using statistical physics to several computer vision problems (Geiger89, Yuille89b, 90abc). It has also been shown that some techniques developed from robust statistics (e.g., M-estimators, the least-trimmed squares estimators) appear naturally within the Baysian formulation by the use of statistical physics approach. In this paper we adapt statistical physics approach to the problem of robust PCA. First we connect some existing PCA learning rules to some energy function (Sec.2). Then in Sec.3 and Sec.4, we generalize the energy function by adding a binary decision field with a given prior distribution so that outliers are taken into consideration. By statistical physics approach, the generalized energy is used to define a Gibbs distribution and to derive an effective energy function which is further used to derive learning rule for robust PCA. The results of comparative experiments have shown that the robust rules proposed here have improved the performances of the existing PCA algorithms significantly.

2 PCA learning rules and energy Function

Assume that $\vec{x}$ is an n-dimensional random vector with zero mean $E\{\vec{x}\} = 0$, the linear combination $\vec{\phi}^T\vec{x}$ is called the principal component if $E\{(\vec{\phi}^T\vec{x})^2\} = Max_{\vec{m}}E\{(\vec{m}^T\vec{x})^2\}$ and $\vec{m}^T\vec{m} = 1$. The solution for the vector $\vec{m}$ is the first dominant eigenvector $\vec{\phi}$ of the data covariance matrix, given by

$$\Sigma\vec{\phi} = \lambda\vec{\phi}, \qquad and \qquad \Sigma = E\{\vec{x}\vec{x}^T\}. \tag{1}$$

where λ is the largest eigenvalue of Σ. The process of finding $\vec{\phi}$ (or λ) is called PCA.

[1]Dr. Lei Xu, G-14 Pierce Hall, Division of Applied Sciences, Harvard University, Cambridge, MA 02138, USA

0-7803-0559-0/92 $3.00

For a given data set $\{\vec{x}_i, i = 1, \cdots N\}$ with zero average $\sum_{i=1}^{N} \vec{x}_i = 0$, a simple approach to do PCA is first to compute simple variance matrix $S = (1/N)\sum_{i=1}^{N} \vec{x}_i \vec{x}_i^T$ and then to solve $S\vec{\phi} = \lambda\vec{\phi}$ to get the largest eigenvalue and eigenvector λ, $\vec{\phi}$ respectively. The simple approach has two major problems. First, it works in *batch* way, which is not suitable for the real applications where data comes incrementally or in the *on line* way. Second, the approach deteriorates its performance drastically and produces unacceptable results in presence of outliers. The first problem can be solved by a number of existing adaptive PCA rules. Among them, the following two are perhaps the most commonly used [1] [2]:

$$\vec{m}_{t+1} = \vec{m}_t + \alpha_t(\vec{x}y - \vec{m}_t y^2), \tag{2}$$

$$\vec{m}_{t+1} = \vec{m}_{t+1} + \alpha_t(\vec{x}y - \frac{\vec{m}_t}{\vec{m}_t^T\vec{m}_t}y^2), \tag{3}$$

where $y = \vec{m}_t^T\vec{x}$ and $\alpha_t \geq 0$ is the learning rate which decreases to zero as $t \to \infty$ with the satisifcation of some conditions, e.g., $\sum_t \alpha_t = \infty$, $\sum_t \alpha^q < \infty$ for some $q > 1$. Due to space limit, this paper will only consider how to extend these two rules into robust versions. However, we like to indicate that many of the existing PCA rules can also be extended in a similar way (see [5] for detail).

It has been shown that the rules eqs.(2& 3) will converge to $\vec{\phi}$ almost surely under some mild conditions[1][2]. By regarding $\vec{m}$ as the synapses weight vector of a linear neuron with output $y = \vec{m}^T\vec{x}$, the rules can be considered as the modifications of the well known Hebbian rule $\vec{m}_{t+1} = \vec{m}_t + \alpha_t\vec{x}y$ with eq.(2) , eq.(3) each having an additional term for preventing $\|\vec{m}_t\|$ goes to ∞ as $t \to \infty$. For each updating, eq.(2) saved the computation of $\vec{m}_t^T\vec{m}_t$ in eq.(3). Moreover, its updating on each component $m_t^{(i)}$ of $\vec{m}_t$ is just based on the locally available variables $m_t^{(i)}$ and y. This kind of locality is usually regarded as "more biologically plausible". The locality has been lost in eq.(3) because $\vec{m}_t^T\vec{m}_t$ is involved for each $m_t^{(i)}$. However, we can show that eq.(3) directly relates to the following energy function eq.(4), which can provide a bridge for generalizing the rules eqs.(2& 3) into robust versions by using statistic physics approach.

$$J(\vec{m}) = \frac{E\{\vec{x}^T\vec{x} - y^2\}}{\vec{m}_t^T\vec{m}_t} = tr\Sigma - \frac{\vec{m}_t^T\Sigma\vec{m}_t}{\vec{m}_t^T\vec{m}_t}, or \quad J(\vec{m}_t) = \frac{1}{N}\sum_{i=1}^{N}(\vec{x}_i^T\vec{x}_i - \frac{\vec{m}_t^T\vec{x}_i\vec{x}_i^T\vec{m}_t}{\vec{m}_t^T\vec{m}_t}) \tag{4}$$

(Note: $J(\vec{m}) \geq 0$, since $\vec{x}^T\vec{x} - \frac{y^2}{\vec{m}_t^T\vec{m}_t} = \|\vec{x}\|^2\sin^2\theta_{xm} \geq 0$, where θ_{xm} is the angle between $\vec{m}_t, \vec{x}$.)

On the one hand, the gradient descent rule for minimizing $J(\vec{m})$ is $\frac{d\vec{m}_t}{dt} = -\frac{\partial J(\vec{m})}{\partial\vec{m}_t} = \frac{2}{\vec{m}_t^T\vec{m}_t}(\Sigma\vec{m}_t - \frac{\vec{m}_t^T\Sigma\vec{m}_t}{\vec{m}_t^T\vec{m}_t}\vec{m}_t)$. On the other hand, by taking expectation on eq.(3) and noticing that $\vec{m}_t$ changes much slower than $\vec{x}$, we have $\vec{m}_{t+1} = \vec{m}_t + \alpha_t(\Sigma\vec{m}_t - \frac{\vec{m}_t^T\Sigma\vec{m}_t}{\vec{m}_t^T\vec{m}_t}\vec{m}_t)$. It is just the discrete form of the gradient descent rule. I.e, eq.(3) is the *adaptive rule* (or called *on line rule* or *stochastic approximation rule*) for minimizing $J(\vec{m})$ in the gradient descent manner.

We further consider eq.(2). As pointed out by Baldi(1991), the rule is not a gradient descent rule of any energy function. However, we can prove the following lemma:

Lemma Let $\vec{h}_1 = \vec{x}y - \vec{m}_t y^2$, $\vec{h}_2 = \vec{x}y - \frac{\vec{m}_t}{\vec{m}_t^T\vec{m}_t}y^2$, then $\vec{h}_1^T\vec{h}_2 \geq 0, E(\vec{h}_1)^T E(\vec{h}_1) \geq 0$.

Proof: (1). $\vec{h}_1^T\vec{h}_2 = (\vec{x}y - \vec{m}_t y^2)^T(\vec{x}y - \frac{\vec{m}_t y^2}{\vec{m}_t^T\vec{m}_t}) = y^2(\|\vec{x}\|^2 - y^2/\|\vec{m}_t\|^2) = y^2\|\vec{x}\|^2\sin^2\theta_{xm} \geq 0$, where θ_{xm} is the angle between $\vec{m}_t, \vec{x}$. (2). $E(\vec{h}_1)^T E(\vec{h}_2) = E(\vec{x}y - \vec{m}_t y^2)^T E(\vec{x}y - \frac{\vec{m}_t y^2}{\vec{m}_t^T\vec{m}_t}) = \vec{m}_t^T\Sigma^2\vec{m}_t - \frac{(\vec{m}_t^T\Sigma\vec{m}_t)}{\vec{m}_t^T\vec{m}_t} = \vec{n}_t^T\vec{n}_t - \frac{(\vec{m}_t^T\vec{n}_t)}{\vec{m}_t^T\vec{m}_t} = \|\vec{n}_t\|^2\sin^2\theta_{mn} \geq 0$, where $\vec{n}_t = \Sigma\vec{m}_t$ and θ_{mn} is the angle between $\vec{m}_t, \vec{n}_t$. **QED.**

Furthermore, we can also show [6] that J only one local (also global) minimum $tr(\Sigma) - \vec{\phi}^T\Sigma\vec{\phi}$, and all the other critical points (i.e., the points satisfy $\frac{\partial J(\vec{m})}{\partial\vec{m}_t} = 0$) are saddle points. Thus, we see that eq.(2) is a down-hill algorithm for minimizing J in both *on line* sense and *average* sense. Since the PC $\vec{\phi}$ is the only

local minimum of J, eq.(2) and eq.(3) will finally reach the same solution. Therefore, we can also connect eq.(2) to J.

3 Robust PCA by statistical physics approach

$J(\vec{m})$ can be regarded as the special cases of the following general energy function:

$$J(\vec{m}) = \frac{1}{N}\sum_{i=1}^{N} z(\vec{x}_i, \vec{m}), \quad z(\vec{x}_i, \vec{m}) \geq 0. \tag{5}$$

where $z(\vec{x}_i, \vec{m})$ is the portion of energy contributed by sample $\vec{x}_i$, and $z(\vec{x}_i, \vec{m}) = (\vec{x}_i^T \vec{x}_i - \frac{\vec{m}_t^T \vec{x}_i \vec{x}_i^T \vec{m}_t}{\vec{m}_t^T \vec{m}_t})$ for J.

Following [7] [8], here we generalize energy eq.(5) into

$$E(\vec{V}, \vec{m}) = NJ(\vec{m}) + E_{prior}(\vec{V}) = \frac{1}{N}\sum_{i=1}^{N} V_i z(\vec{x}_i, \vec{m}) + E_{prior}(\vec{V}) \tag{6}$$

where $\vec{V} = \{V_i, i = 1, \cdots, N\}$ is a binary field $\{V_i\}$ with each V_i being random variable which takes value either 0 or 1. V_i acts as a decision indicator for deciding whether $\vec{x}_i$ is an outlier or a sample. When $V_i = 1$, the portion of energy contributed by sample $\vec{x}_i$ is taken into consideration; otherwise, it is equivalent to discarding $\vec{x}_i$ as an outlier. $E_{prior}(\vec{V})$ is the priori portion of energy contributed by the priori distribution of $\{V_i\}$. A natural choice is $E_{prior}(\vec{V}) = \lambda\sum_{i=1}^{N}(1 - V_i)$, which has a natural interpretation: for fixed $\vec{m}$ it is energetically favourable to set $V_i = 1$ (i.e., not regarding $\vec{x}_i$ as an outlier) if $z(\vec{x}_i, \vec{m}) < \sqrt{\lambda}$ (i.e., the portion of energy contributed by $\vec{x}_i$ is smaller than a prespecified threshold) and to set it to 0 otherwise.

The goal is to minimize $E[\vec{V}, \vec{m}]$ with respect to $\{V_i\}$ and $\vec{m}$ simultaneously in the constraint that every V_i only takes binary value. This problem is a mixture of discrete and continuous optimizations. The solution usually can not be calculated analytically, and is also hard to be obtained by gradient descent approach.

To help us solve this problem, based on statistical physics we define a Gibbs distribution (Parisi 1988)[9]:

$$P[\vec{V}, \vec{m}] = \frac{1}{Z}e^{-\beta E[\vec{V}, \vec{m}]}, \tag{7}$$

where Z is the partition function which ensures $\sum_{\vec{V}}\sum_{\vec{m}} P[\vec{V}, \vec{m}] = 1$. Now minimizing $E[\vec{V}, \vec{m}]$ is equivalent to maximizing $P[\vec{V}, \vec{m}]$. But this still did not get rid of the difficulty of discrete optimization with respect to binary value $\{V_i\}$. One solution for the problem is to compute the marginal probability distribution $P_{margin}(\vec{m})$ by averaging out the variables $\{V_i\}$ in the consideration of the constraint that they only take binary values, and then to use the maximization of $P_{margin}(\vec{m})$ to approximate the maximization of $P(\vec{V}, \vec{m})$. Analytically, $P_{margin}(\vec{m})$ can be computed as follows:

$$\begin{aligned} P_{margin}(\vec{m}) &= \frac{1}{Z}\sum_{\vec{V}} e^{-\beta\sum_i\{V_i z(\vec{x}_i,\vec{m})+\lambda(1-V_i)\}} = \frac{1}{Z}\prod_i \sum_{V_i=\{0,1\}} e^{-\beta\{V_i z(\vec{x}_i,\vec{m})+\lambda(1-V_i)\}} \\ &= \frac{1}{Z}\prod_i\{e^{-\beta\lambda} + e^{-\beta z(\vec{x}_i,\vec{m})}\} = \frac{e^{-N\beta\lambda}}{Z}\prod_i\{1 + e^{-\beta\{z(\vec{x}_i,\vec{m})-\lambda\}}\}. \end{aligned} \tag{8}$$

Defining $Z_m = Ze^{N\beta\lambda}$, we obtain

$$P_{margin}(\vec{m}) = \frac{1}{Z_m}e^{-\beta E_{eff}(\vec{m})}, \quad E_{eff}(\vec{m}) = \frac{-1}{\beta}\sum_i \log\{1 + e^{-\beta\{z(\vec{x}_i,\vec{m})-\lambda\}}\}. \tag{9}$$

Maximizing $P_{margin}(\vec{m})$ with respect to x is equivalent to minimizing $E_{eff}(\vec{m})$. The form of E_{eff} can be regarded as a generalization of a robust redesecending M-estimators (Huber 1981) to PCA problem. It is clear that each term in the sum for E_{eff} is just $z(\vec{x}_i, \vec{m})$ when it has a small value but becomes constant as

$z(\vec{x}_i, \vec{m}) \rightarrow \infty$. In this way, outliers which are more likely to yield large $z(\vec{x}_i, \vec{m})$ are considered differently from samples, and thus the estimation $\vec{m}$ obtained by minimizing $E_{eff}(\vec{m})$ will be robust in resisting outlier.

$E_{eff}(\vec{m})$ is usually not a convex function and may have many local minima. The statistical physics framework suggests using deterministic annealing to minimize $E_{eff}(\vec{m})$. That is, by the gradient descent approach to minimize $E_{eff}(\vec{m})$ for small β and then tracking the minimum as β increases to infinity (the zero temperature limit):

$$\frac{\partial \vec{m}_t}{\partial t} = -\frac{\partial E_{eff}(\vec{m})}{\partial \vec{m}_t}, \quad or \ \vec{m}_{t+1} = \vec{m}_t - \alpha \frac{\partial E_{eff}(\vec{m})}{\partial \vec{m}_t} = \vec{m}_t - \alpha \sum_i \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}} \frac{\partial z(\vec{x}_i, \vec{m})}{\partial \vec{m}_t}. \tag{10}$$

Specifically, for J we have the following *batch* way rule for PCA:

$$\vec{m}_{t+1} = \vec{m}_{t+1} + \alpha \sum_i \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}} (\vec{x}_i y_i - \frac{\vec{m}_t}{\vec{m}_t^T \vec{m}_t} y_i^2), \tag{11}$$

where as $t \rightarrow \infty$, the learning rate $\alpha \rightarrow 0$ and the annealing parameter $\beta \rightarrow \infty$.

Finally, the converged vector $\vec{m}_{converge}$ is taken as the resulted principal component vector which has avoided the affects of outliers. In addition, a by-product can be easily obtained by

$$V_i = 1, \quad if \ z(\vec{x}_i, \vec{m}) > \lambda \ and \ V_i = 0, \ otherwise. \tag{12}$$

which indicates whether $\vec{x}_i$ has been considered as an outlier depending on whether $V_i = 0$ or 1.

4 Robust adaptive PCA rules and computer experiments

The adaptive versions of eq.(10 & 11) are quite simple to get: just to move away the summation in eq.(10 & 11) or equivalently to minimize the following energy portion contributed by the present sample $\vec{x}_i$:

$$e(\vec{m}, \vec{x}_i) = \frac{-1}{\beta} \log\{1 + e^{-\beta\{z(\vec{x}_i, \vec{m}) - \lambda\}}\}. \tag{13}$$

By gradient descent approach, we get

$$\vec{m}_{t+1} = \vec{m}_t - \alpha \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}} \frac{\partial z(\vec{x}_i, \vec{m})}{\partial \vec{m}_t}, \quad \vec{m}_{t+1} = \vec{m}_{t+1} + \alpha \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}} (\vec{x}_i y_i - \frac{\vec{m}_t}{\vec{m}_t^T \vec{m}_t} y_i^2). \tag{14}$$

Under the assumption that $\vec{x}_i$ comes from a stationary process and that $\vec{m}_t$ changes much slower than $\vec{x}_i$, it is not difficult to see that eq.(14) is really an adaptive or stochastic approximation rule which minimizes $E_{eff}(\vec{m})$ of eq.(9) in the gradient descent manner.

We can observe that the difference between eq.(3) and eq.(14) in that the learning rate α has been modified by a multiplicative factor $\alpha_m = \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}}$, which adaptively modifies the learning rate to suit the current input $\vec{x}_i$. This modifying factor takes a similar role as that does in an algorithm [4] proposed by one of the present authors for robust line fitting. Based on the connection between eq.(2) and energy function J (discussed at the end of Sec.2), we can also formally use the modifying factor α_m to turn the rule eq.(2) into the following robust version:

$$\vec{m}_{t+1} = \vec{m}_{t+1} + \alpha \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}} (\vec{x}_i y_i - \vec{m}_t y_i^2), \tag{15}$$

and the corresponding *batch way* rule is given as follows

$$\vec{m}_{t+1} = \vec{m}_{t+1} + \alpha \sum_i \frac{1}{1 + e^{\beta(z(\vec{x}_i, \vec{m}) - \lambda)}} (\vec{x}_i y_i - \vec{m}_t y_i^2), \tag{16}$$

In the rest of this section, we introduce some results obtained from comparative experiments on the rules given in sec.2 and their robust versions given the above.

Let $\vec{x}$ be a 3-D vector coming from a 3-D population of 400 samples with zero mean. These samples locate on a ring in R^3 space. Its projections on $x-y$, $y-z$ and $z-x$ planes are shown in Fig.1.

Without the presence of outliers, on this data set the simple batch approach (see the beginning of sec.2) finds that the PC vector is $\phi_p = [0.0710, 0.8876, 0.4551]^T$. The results of using eq.(2 & 3) are given in Fig.3. The two rules behaves almost identically and converge to ϕ_p quite perfectly, where the vertical axis θ denotes the angle between the present m_t and ϕ_p.

Then the data set is contaminated by 10 outlier points. The projection of this spoiled data set on $x-y$, $y-z$ and $z-x$ planes are shown in Fig.2 (note: the scaling changed here, this is why the shapes also changed). This time, the result of the simple batch approach is $\phi_r = [-0.8285, 0.1567, 0.5375]^T$ and the angle between ϕ_r and the right one ϕ_r is 71.04 degrees. It is obviously that the result is unacceptable at all. This revealed that the simple batch way has no outlier-resisting ability.

Fig.4 gives the results of using the robust rules eq.(15), eq.(14) and the following *batch way* versions of the original rules eq.(2) and eq.(3):

$$\vec{m}_{t+1} = \vec{m}_t + \alpha_t \sum_i (\vec{x}_i y_i - \vec{m}_t y_i^2), \tag{17}$$

$$\vec{m}_{t+1} = \vec{m}_{t+1} + \alpha_t \sum_i (\vec{x}_i y_i - \frac{\vec{m}_t}{\vec{m}_t^T \vec{m}_t} y_i^2), \tag{18}$$

In Fig.4, (also in Fig.5), the solid curves express the learning curves by the unnormalized rules eq.(2), eq.(17), eq.(16) and eq.(15); while the dotted curves express the learning curves by the normalized rules eq.(3), eq.(18), eq.(11) and eq.(14). Moreover, "R" denotes the results of the robust rules eq.(16), eq.(15),eq.(11) and eq.(14); "U" denotes the results of the unrobust rules eq.(2), eq.(17), eq.(3) and eq.(18).

It can be seen that the learning process by eq.(17) and eq.(18) converge identically to a vector which has an angle of 20.97 degree to the correct ϕ_p. The result is certainly much better than that obtained by the sample batch approach. This means that the rules eq.(17) and eq.(18) do have some outlier-resisting ability. However, this solution has still a quite big error. Favorably, the learning process by eq.(16) and eq.(11) each converged to a vector which has an angle of only about 1.06 degrees or 1.30 degrees respectively to the correct ϕ_p. These solutions are so accurate and significantly better than those obtained by eq.(17) and eq.(18). This shows that the robust rules eq.(16) and eq.(11) do resist outliers very well. In this experiment, the parameters used are $\alpha = 0.3, \beta = 0.5$ and $\lambda = 4$ for all the rules. Here, for simplicity we kept these parameters in constant. It could be expected that better solutions could be obtained when $\alpha \to 0, \beta \to \infty, \lambda \to 0$ in a suitable way.

The results of using the *on line* rules eq.(2) and eq.(3) and their corresponding robust versions eq.(15) and eq.(14) are shown in Fig.5. Similarly, we again see that the learning process by eq.(2) and eq.(3) converge identically to a vector which has an angle of again about 21 degrees to the correct ϕ_p; while the learning process by eq.(15) and eq.(14) each converged to a very accurate solution which has an angle of only about 0.36 degree or 4.04 degrees respectively to the correct ϕ_p. This again revealed that the robust rules proposed in this paper have improved the outlier-resisting ability of the conventional PCA algorithms significantly. In this example, the parameters used are $\alpha = 0.001, \beta = 0.5$ and $\lambda = 4$ respectively.

We can also observe that the normalized and the unnormalized versions of unrobust rules always performs almost identically. But for the robust rules, the unnormalized versions converge faster than the the normalized ones do. In addition, the unnormalized versions can save the computation of $m_t^T m_t$ at each updating. Thus, we suggest that eq.(16) and eq.(15) are better choices in practice. However, eq.(11) and eq.(14) are also of importance theoretically because their direct connections to the energy function J.

6. Conclusion

Statistical physics approach has been adapted to the problem of robust PCA. The robust PCA learning rules have been proposed by generalizing the two commonly used Oja PCA rules, which are shown to perform badly in the presence of outliers. The results of comparative experiments have shown that the proposed robust rules have improved the performances of the two Oja rule significantly.

References
[1].E.Oja, J. Math. Biology, **16**, 1982, 267-273. [2]. E.Oja and J.Karhunen, J. Math. Anal. Appl. **106**, 1985, 69-84. [3]. L.Xu, *Proc. of IJCNN 1991-Singapore*, Nov., 1991, pp2368-2373. [4]. L.Xu, E.Oja and C.Y.Suen, Modified Hebbian learning for curve and surface fitting. Neural Networks, 1992, in press. [5]. L.Xu and A.L.Yuille, Robust principal component analysis by self-organizing rules based on statistical physics approach, to be submitted, 1992. [6]. L.Xu, Self-organizing rules for statistical analysis of orthogonal components: a comprehensive study, to be submitted, 1992. [7].A.L.Yuille, Neural computation **2**, 1990, 1-24. [8]. A.L.Yuille, T. Yang and Davi Geiger, Robust statistics, transparency and correspondence, Harvard Robotics Lab. Tech. Rep. 90-7, 1990. [9]. G.Parisi, *Statistical Field Theory*, Addison-Welsley, 1988.

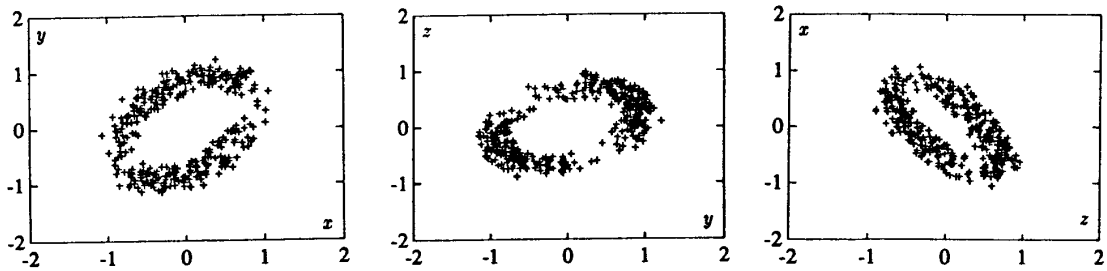

Fig.1 the projections of data set on x-y, y-z, z-x planes (without outliers)

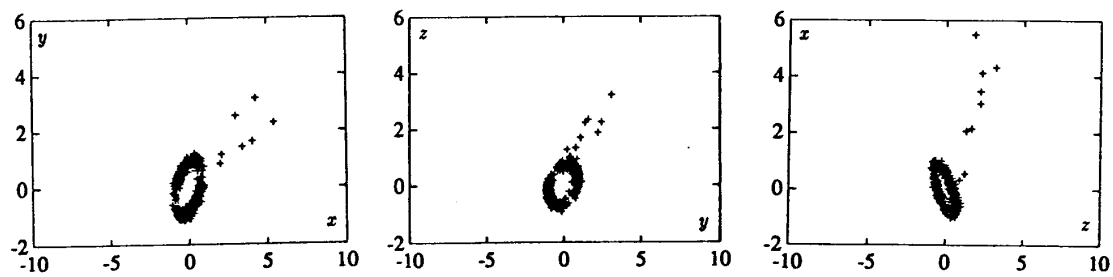

Fig.2 the projections of data set on x-y, y-z, z-x planes (with outliers)

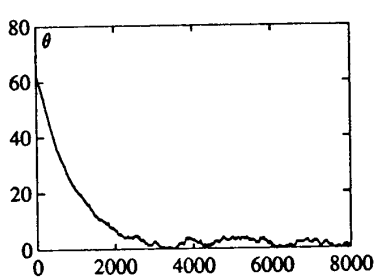

Fig.3 learning on the data set without outliers

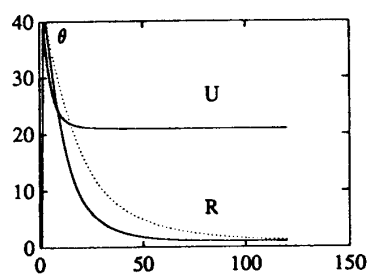

Fig.4 learning on the data set with outliers: batch way

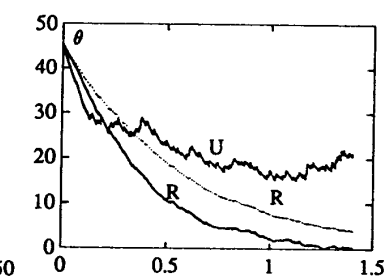

Fig.5 learning on the data set with outliers: on line way