Multisets Modeling Learning: An Unified Theory for Supervised and Unsupervised Learning

(Invited Paper)

Lei Xu

The Chinese University of Hong Kong and Peking University

Abstract— An unified theory is proposed for putting together supervised learning and unsupervised learning (including clustering, PCA-type selforganizing and topological map) into one single frame. By this theory, different special cases will automatically lead us to supervised learning for feedforward networks and for modular architecture of local experts, to various types of unsupervised learning including data clustering, PCA and k-principal components analysis (k-PCA), minor component analysis (MCA) and k-minor components analysis (k-MCA), principal subspace analysis (PSA) and minor subspace analysis (MSA), as well as their extensions to the localized versions (e.g., local PCA, local MCA, ..., etc.) Furthermore, it is also shown that the theory can be extended to cover self-organizing topological map.

I. Introduction

Supervised and unsupervised learnings are two major branches in neural network learning. Each has been extensively studied. The readers are referred to [16] for a recent survey on supervised learning and to [1] for a recent survey on unsupervised learning.

Supervised and unsupervised learnings are developed from different theories, and usually regarded being essentially different. Even for unsupervised learning itself, we have three types: clustering, PCA-type selforganizing, and topological map. They are also developed from different motivations and based on different theories or heuristics, and considered to be considerably different from each other too.

This paper proposes a so called *Multisets Modeling Learning theory*. It is an unified theory for putting together supervised learning and unsupervised learning (including clustering, PCA-type selforganizing and topological map) into one single frame. Under this theory, a general set specified by a given property is used as the representation form of a model and a number of sets are used to represents multiple models. The error of a data point $\vec{x}$ to a model is defined as the minimum one among the errors of $\vec{x}$ to every elements in the model set. Given a data set, the purpose of learning is to determine the property of each model set. We consider the cases that each such property is specified by some parametric equation. In this case, the learning is to determine the parameters of equation. By this theory, we are lead to supervised learning for the conventional feedforward networks [16] when the parametric equation is an explicit parametric function, and particularly to supervised learning for modular architecture of local experts [4] when we have several sets with each being specified by an explicit parametric function; we are lead to unsupervised learning when the model sets are specified by points, lines, hyperplanes, hypercurves, hypersurfaces, subspaces, and manifolds. Particularly, we get *center location* for a point and *clustering* for several points; we get the *standard PCA* for a line and its extension—*local PCA* for several lines; we get *MCA* for a hyperplane and *local MCA* for several hyperplanes; as well as we get *k-PCA, PSA, k-MCA, MSA* for one subspace or linear manifold and their localized extensions for several subspaces or linear manifolds. Furthermore, we will also show that the theory can be extended to cover self-organizing topological map[7].

The author's address: Dept. of Computer Science, HSH ENG BLDG, Room 1006, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong.
I would like to acknowledge partial support from CUHK by a DIRECT GRANT for Research 93-94.

II. Multisets Modeling Learning Theory

We propose a very general representation for describing a model. This representation is a *set* specified by

$$M(\vec{x}) = \{\vec{x} : p(\vec{x})\}, \qquad (1)$$

where $p(\vec{x})$ denote a general predicate proposition, which can be any of the following possibilities:

- It is described a single implicit function defined by $F(\vec{x}) = 0$. That is, $p(\vec{x}) = true$ when $\vec{x}$ is a root of the equation $F(\vec{x}) = 0$. $F(\vec{x})$ can be either linear or nonlinear on D_x, where D_x is the domain of $\vec{x}$.
- It is described by an explicit function $\vec{\eta} = f(\vec{\xi})$ with $\vec{x} = [\vec{\xi}^{\,t}, \vec{\eta}^{\,t}]^T$. That is, $p(\vec{x}) = true$ when this $\vec{x}$ let $\vec{\eta} = f(\vec{\xi})$ hold. Again, the $f(\vec{\xi})$ is a general function, which can be either linear or nonlinear.
- It is described by an inequality $F(\vec{x}) > 0$ or $F(\vec{x}) < 0$. That is, $p(\vec{x}) = true$ when this inequality holds.
- It consists of a number of propositions $p_1(\vec{x}), \cdots, p_r(\vec{x})$ which are combined together by logic connectives $\wedge, \vee, \neg$. Each of these individual propositions can be either of the above three cases. E.g., we can have $p(\vec{x}) = p_1(\vec{x}) \wedge p_2(\vec{x}) \vee p_3(\vec{x})$, with $p_1(\vec{x})$ described by $F(\vec{x}) = 0$, $p_2(\vec{x})$ by $\vec{\eta} = f(\vec{\xi})$ and $p_3(\vec{x})$ by $F(\vec{x}) > 0$.
- It can also be a proposition or a logic combination of several propositions that are described by languages but not by the above mathematical expressions.

The first two cases are equivalent to the models usually used in the conventional studies of modeling problems. The last three generalize the conventional descriptions of models. By such generalizations, a model can also be an area or a volume with any shape, a combination of areas or volumes, or even be an arbitrary subset on the domain D_x.

With this model, a point $\vec{x}_i \in D_x$ is said to satisfy the model $M(\vec{x})$ if $\vec{x}_i \in M(\vec{x})$. If $\vec{x}_i$ does not satisfy $M(\vec{x})$, we define the following error

$$\varepsilon^q(\vec{x}_i) = \min_{\vec{y} \in M} |\vec{x}_i - \vec{y}|^q, \; p \geq 1 \qquad (2)$$

for its discrepancy from $M(\vec{x})$, where $|\vec{u}|^q = \sum_{i=1}^n |u_i|^q$, for $\vec{u} = [u_1, \cdots, u_n]^T$. Specifically, when $q = 2$, we get the square error (or sometime called distance) between the point $\vec{x}_i$ and the set $M(\vec{x})$; when $q = 1$, we get the L_1 error for $\vec{x}_i$ from $M(\vec{x})$. Both the errors are particularly useful in practice.

Given a data set $\mathcal{D}_x = \{\vec{x}_1, \cdots, \vec{x}_N\}$, the problem of our learning is to specify $p(\vec{x})$ in eq.(1) such that the sum $\sum_{i=1}^N \varepsilon^q(\vec{x}_i) = \sum_{i=1}^N \min_{\vec{y} \in M} |\vec{x}_i - \vec{y}|^q$ is minimized.

In this paper, we concentrate on the cases that $p(\vec{x})$ can be determined by a set of parameters W. In thes cases, we denote the model by $M(\vec{x}, W)$ and the learning problem becomes the problem of minimizing J^q with respect to W:

$$J^q = \sum_{i=1}^{N} e^q(\vec{x}_i, W) = \sum_{i=1}^{N} \min_{\vec{y} \in M(\vec{x}, W)} |\vec{x}_i - \vec{y}|^q, \quad (3)$$

The further implementation of this minimization will depend on the specific form of $M(\vec{x}, W)$ and will be discussed in the following sections.

More generally, data $\mathcal{D}_x$ may have several modes, or in the other words, $\mathcal{D}_x$ is a mixture of data from several different objects or models. We propose to use a number of models $M_m(\vec{x}, W_m), m = 1, \cdots, M$ to deal with these cases. The learning problem is now specified by the minimization of the mixture error with respective to parameters $W_1, \cdots, W_M$:

$$J_M^q = \sum_{m=1}^{M} \sum_{i=1}^{N} \omega_m(\vec{x}_i) e^q(\vec{x}_i, W_m)$$
$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \omega_m(\vec{x}_i) \min_{\vec{y} \in M_m(\vec{x}_i, W_m)} |\vec{x}_i - \vec{y}|^q$$

s.t. $\sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1.$, where $1 \geq \omega_m(\vec{x}_i) \geq 0$ is a weighting coefficient on the k-th model. It can be regarded as the probability of $\vec{x}_i$ generated from $M_m(\vec{x}, W_m)$. The set of NM such coefficients $\omega_m(\vec{x}_i)$'s are usually unknown to us, and need to be determined through the above minimization. As a result, the over-sized unknowns make the minimization undetermined. To avoid this difficulty, we minimize a new cost function

$$J_M^q = \sum_{m=1}^{M} \sum_{i=1}^{N} \omega_m(\vec{x}_i) e^q(\vec{x}_i, W_m)$$
$$+\beta \sum_{m=1}^{M} \sum_{i=1}^{N} \omega_m(\vec{x}_i) \ln \omega_m(\vec{x}_i), \ s.t. \ \sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1. \quad (4)$$

Where β is a given constant. The new term is to force $\omega_m(\vec{x}_i)$ close to either 0 or 1, and can also be interpreted as the entropy of distribution $\omega_m(\vec{x}_i)$ [3] [22] [9].

The interesting point is that this minimization can be implemented by two iterative steps:

- With the parameters $W_m^{(k)}, m = 1, \cdots, M$ fixed, considering the constraint $\sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1$ and by $\nabla_{\omega_m(\vec{x}_i)} J_M^q = 0$, we can get
  $$e^q(\vec{x}_i, W_m^{(k)}) + \lambda + \beta(\ln \omega_m(\vec{x}_i) + 1) = 0, \text{ or}$$
  $$\omega_m^{(k)}(\vec{x}_i)^{(k)} = \tfrac{1}{Z} e^{-\beta^{-1} e^q(\vec{x}_i, W_m^{(k)})}, \quad Z = e^{-\beta^{-1}\lambda - 1}.$$
  Furthermore, we have
  $$\sum_{m=1}^{M} \omega_m^{(k)}(\vec{x}_i) = \sum_{m=1}^{M} \tfrac{1}{Z} e^{-\beta^{-1} e^q(\vec{x}_i, W_m^{(k)})} = 1$$
  and $Z = 1/\sum_{m=1}^{M} e^{-\beta^{-1} e^q(\vec{x}_i, W_m^{(k)})}$, which leads to
  $$\omega_m^{(k)}(\vec{x}_i) = \frac{e^{-\beta^{-1} e^q(\vec{x}_i, W_m^{(k)})}}{\sum_{m=1}^{M} e^{-\beta^{-1} e^q(\vec{x}_i, W_m^{(k)})}}. \quad (5)$$
- With the parameters $\omega_m^{(k)}(\vec{x}_i)$'s fixed, the minimization of J_M^q with respect to $W_m, m = 1, \cdots, M$ can be decomposed into each of the independent minimizations with respect to W_m
  $$J_{M_m}^q(W_m) = \sum_{i=1}^{N} \omega_m^{(k)}(\vec{x}_i) e^q(\vec{x}_i, W_m), \quad (6)$$
  which are just the weighted version of eq.(3) and can be solved in the same way as the single model case. The result will give the updated $W_m^{(k+1)}, m = 1, \cdots, M$.

Given the initial $W_m^{(0)}, m = 1, \cdots, M$, we can repeatedly implement the above two steps. The iteration will converge to at least a local minimum of J_M^q as long as the second step let $J_{M_m}^q(W_m^{(k+1)}) \leq J_{M_m}^q(W_m^{(k)}), m = 1, \cdots, M$ with $J_{M_m}^q(W_m^{(k+1)}) \neq J_{M_m}^q(W_m^{(k)})$ for a number of iterations. The reason is that each step is actually doing a descent search (not gradient descent search) of J_M^q. In the following sections, under the specific forms of $M_m(\vec{x}, W_m)$ we will show that this two step method is closely related to the well known EM algorithm developed under the incomplete data theory[2].

III. Supervised Learning

For a supervised learning problem, $\vec{x}$ consists of two parts[1] $\vec{x} = [\vec{\xi}, \vec{\eta}]$, and the $p(\vec{x})$ in eq.(1) is now specified by an explicit function $\vec{\eta} = f(\vec{\xi}, W)$. Correspondingly, the set $M(\vec{x}, W)$ can also been written as the Cartesian product

$$M(\vec{x}, W) = M(\vec{\xi}) \times M(\vec{\eta}) = D_\xi \times M(\vec{\eta})$$

where D_ξ is the domain of $\vec{\xi}$, and

$$M(\vec{\eta}) = \{\vec{\eta} : \vec{\eta} = f(\vec{\xi}, W), \forall \vec{\xi} \in D_\xi\}.$$

It follows from eq.(2) that

$$e^q([\vec{\xi}, \vec{\eta}]) = \min_{\vec{\eta}' \in M(\vec{\eta})} \|[\vec{\xi}, \vec{\eta}_i] - [\vec{\xi}, \vec{\eta}_i']|^q = |\vec{\eta}_i - f(\vec{\xi}, W)|^q.$$

Given a data set (or called the training set) $\mathcal{D}_x$ which is now usually written in the set of input-output pairs $\mathcal{D}_x = \mathcal{D}_{[\vec{\xi}, \vec{\eta}]} = \{[\vec{\xi}_1, \vec{\eta}_1], \cdots, [\vec{\xi}_N, \vec{\eta}_N]\}$. The problem of eq.(3) now becomes the minimization of the following J^q with respect to the parameters W

$$J^q = \sum_{i=1}^{N} |\vec{\eta}_i - f(\vec{\xi}, W)|^q, \quad (7)$$

Obviously, it is just the conventional least L_p error supervised learning. When $q = 2$ and $f(\vec{\xi}, W)$ being a feedforward network with input $\vec{\xi}$, Backpropagation technique can be used to implement this learning. In addition, all the other variants of Backpropagation[16] can also be used to solve eq.(7).

Furthermore, for the cases that data comes from a mixture of multiple models, we have that each of model sets $M_m(\vec{x}, W_m), m = 1, \cdots, M$ is specified by $f_m(\vec{\xi}, W_m), m = 1, \cdots, M$, and that the learning problem eq.(4) becomes the problem of minimizing

$$J_M^q = \sum_{m=1}^{M} \sum_{i=1}^{N} \omega_m(\vec{x}_i) |\vec{\eta}_i - f_m(\vec{\xi}, W_m)|^q$$
$$+\beta \sum_{m=1}^{M} \sum_{i=1}^{N} \omega_m(\vec{x}_i) \ln \omega_m(\vec{x}_i), \ s.t. \ \sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1; \quad (8)$$

and eqs.(5)(6) become

$$\omega_m^{(k)}(\vec{x}_i) = \frac{e^{-\beta^{-1} |\vec{\eta}_i - f_m(\vec{\xi}, W_m^{(k)})|^q}}{\sum_{m=1}^{M} e^{-\beta^{-1} |\vec{\eta}_i - f_m(\vec{\xi}, W_m^{(k)})|^q}}. \quad (9)$$

$$J_{M_m}^q(W_m) = \sum_{i=1}^{N} \omega_m^{(k)}(\vec{x}_i) |\vec{\eta}_i - f_m(\vec{\xi}, W_m)|^q, \quad (10)$$

This can be regarded as a generalization of supervised learning for a single network to a modular architecture consisting of several networks. When $q = 2$, this learning is a special case of the supervised learning for *adaptive mixtures of local experts*[4] [5] [6] [18]. In addition, eq.(10) also provides an alternative extension of the *adaptive mixtures of local experts* from L_2 error (Guassian) to nonGaussian L_p error.

[1]Strictly, we should write as $\vec{x} = [\vec{\xi}\,^t, \vec{\eta}\,^t]^T$. Here we omit the transposition for simplicity.

IV. Unsupervised Learning: Single Model

We consider the cases of model eq.(1) that $p(\vec{x})$ is specified by a point, a line, a hyperplane, a subspace, a linear manifold, and show how the proposed theory perform various unsupervised learning tasks including PCA, k-PCA, MCA, k-MCA, PSA and MSA as well as new unsupervised learning tasks.

A. A point: location of mode

When $p(\vec{x})$ is specified by a parametric point $\vec{a}$. The learning problem eq.(3) will become the simpliest case—the minimization of $J^q = \sum_{i=1}^N |\vec{x}_i - \vec{a}|^q$.

This is the simplest unsupervised learning task though it is often ignored due to its simplicity. It learns the location of the mode or center of data D_x. When $q = 2$, it produces the mean vector of D_x, i.e., $\vec{a} = \frac{1}{N}\sum_{i=1}^N \vec{x}_i$.

B. A line: PCA

First we consider the lines passing through the origin of the coordinate system. In this case, any line can be represented by a vector $\vec{w}$. That is, $p(\vec{x})$ is specified by a vector $\vec{w}$ and we have

$M(\vec{x}) = \{\vec{x} : \vec{x} = c\vec{w},\ c\ is\ an\ arbitrary\ constant\}$

when $q = 2$, the error eq.(2) becomes

$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = |\vec{x} - \vec{w}\vec{w}\,{}^t\vec{x}|^2,$

and thus the learning problem eq.(3) becomes the minimization of

$$J^2 = \sum_{i=1}^N |\vec{x}_i - \vec{w}\vec{w}\,{}^t\vec{x}_i|^2, \qquad (11)$$

This is actually the special case of the LMSER criterion proposed in [13] [17] for a single linear neuron. From the studies of [13] [17], we know that the minimization of J_2 will let $\vec{w}$ be the eigenvector of R that corresponds to the largest eigenvalue, where $R = \sum_{i=1}^N \vec{x}_i\vec{x}_i^{\,t}$ is the correlation matrix of $\vec{x}$. In addition, if considering a vector $\vec{w}$ of unit length at the beginning, we can also get

$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = |\vec{x} - \vec{w}\vec{w}\,{}^t\vec{x}|^2_{|\vec{w}|^2=1}$

and an equivalent version of eq.(11) by

$$J^2 = \sum_{i=1}^N |\vec{x}_i - \vec{w}\vec{w}\,{}^t\vec{x}_i|^2_{|\vec{w}|^2=1}, \qquad (12)$$

It can be shown that this version performs in the same way as eq.(11).

When the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$, R becomes the covariance matrix, then the above learnings perform PCA, a problem studied widely in the neural network literature [10][11] [17] [21].

Next we consider the lines passing through any point $\vec{a}$. In this case, any such line can be represented by two vectors $\vec{w}, \vec{a}$ such that

$M(\vec{x}) = \{\vec{x} : \vec{x} - \vec{a} = c\vec{w},\ c\ is\ an\ arbitrary\ constant\}$

and we have

$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = |(\vec{x} - \vec{a}) - \vec{w}\vec{w}\,{}^t(\vec{x} - \vec{a})|^2.$

The eq.(11) will becomes

$$J^2 = \sum_{i=1}^N |(\vec{x}_i - \vec{a}) - \vec{w}\vec{w}\,{}^t(\vec{x}_i - \vec{a})|^2. \qquad (13)$$

This minimization will lead to PCA even when we do not have the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$. Here, the estimation of mean by $\vec{a}$ and the search of principal component by $\vec{w}$ are processed simultaneously. Although we can also use the existing PCA learning rules [10][11] [17] [21] to perform PCA in this case, we need first to substruct the mean from each $\vec{x}_i$ and then start to learn the principal component by $\vec{w}$.

Similar to the case of obtaining eq.(12), we can also get an equivalent version for eq.(13).

C. A hyperplane: MCA

First we consider the planes passing through the origin of the coordinate system. In this case, $p(\vec{x})$ is specified by a hyperplane $\vec{w}\,{}^t\vec{x} = 0$. When $q = 2$, the error eq.(2) becomes

$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = (\vec{w}\,{}^t\vec{x})^2/|\vec{w}|^2,$

or $\quad\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = (\vec{w}\,{}^t\vec{x})^2_{|\vec{w}|^2=1},$

and the model problem eq.(3) becomes the minimization of

$$J^2 = \sum_{i=1}^N \frac{(\vec{w}\,{}^t\vec{x}_i)^2}{|\vec{w}|^2},\ or\ J^2 = \sum_{i=1}^N (\vec{w}\,{}^t\vec{x}_i)^2_{|\vec{w}|^2=1}. \qquad (14)$$

From $\nabla_{\vec{w}}\, J^2 = 0$, we have

$$R\vec{w} - \frac{\vec{w}\,{}^tR\vec{w}}{\vec{w}\,{}^t\vec{w}}\vec{w} = 0,\ or\ R\vec{w} - \lambda\vec{w} = 0, \qquad (15)$$

where R is the correlation matrix of $\vec{x}$. All the solutions of eq.(15) are the eigenvectors of R. Also only the one corresponding to the smallest eigenvalue make J^2 arrive it minimum, and all the other solutions are the saddle points of J^2. When the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$, R becomes the covariance matrix, then this learning performs MCA [14] [12]. In fact, this learning is similar to the MCA learning proposed by [14] for curve and surface fitting. So the detail studies given in [14] apply to here too.

For a hyperplane passing through any point $\vec{a}$, it can be represented by an equation $(\vec{x} - \vec{a})\,{}^t\vec{w} = 0$. For $q = 2$, the error eq.(2) becomes

$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = \frac{[\vec{w}\,{}^t(\vec{x}-\vec{a})]^2}{|\vec{w}|^2}$

or $\quad\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = (\vec{w}\,{}^t(\vec{x} - \vec{a}))^2_{|\vec{w}|^2=1}$

The eq.(14) will become

$$J^2 = \sum_{i=1}^N \frac{[\vec{w}\,{}^t(\vec{x}_i - \vec{a})]^2}{|\vec{w}|^2},\ or\ J^2 = \sum_{i=1}^N [\vec{w}\,{}^t(\vec{x}_i - \vec{a})]^2_{|\vec{w}|^2=1}, \qquad (16)$$

This learning will perform MCA even when $\frac{1}{N}\sum_{i=1}^N \vec{x}_i \neq 0$. Now, the learning of mean by $\vec{a}$ and the search of minor component by $\vec{w}$ are processed simultaneously. However, for the rule given in [14] we need first to substruct the mean from each $\vec{x}_i$ and then start to learn the minor component by $\vec{w}$.

D. A subspace and a linear manifold: k-PCA, PSA, k-MCA, and MCA

Assume that a subspace S is spanned by a set of unit length orthogonal vectors $\vec{w}_1, \cdots, \vec{w}_k$, where these vectors are of the same dimension as $\vec{x}$. Thus, we have $W^TW = I$ for $W = [\vec{w}_1, \cdots, \vec{w}_k]$. Let $p(\vec{x})$ is specified by this subspace, i.e., $M(\vec{x}) = \{\vec{x} : \vec{x} \in S\}$.

When $q = 2$, the error eq.(2) actually represents the perpendicular distance of $\vec{x}_i$ to S, which is given by $|\vec{x}_i - P\vec{x}_i|^2$. $P\vec{x}_i$ is the orthogonal projection of $\vec{x}_i$ on S and

$$P = W(W^TW)^{-1}W^T = WW^T$$

is called the orthogonal projector to S. Thus, the error eq.(2) becomes

$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = |\vec{x} - WW^T\vec{x}|^2_{W^TW=I}$

and thus the learning problem eq.(3) will become the minimization of

$$J^2 = \sum_{i=1}^{N} |\vec{x}_i - WW^T\vec{x}_i|^2_{W^TW=I}. \qquad (17)$$

From $\nabla_W J^2 = 0$, we have $RW - PRW + RW - RPW + 2W\Lambda = 0$, where R is the correlation matrix of $\vec{x}$. Since $P = WW^T, W^TW = I$, we further have

$$RW = W(2\Lambda - W^TRW)$$

The solutions are the matrices of W that consists of eigenvectors of R. Also only the one corresponding to the k largest eigenvalues make J^2 arrive its minimum, and all the other solutions are the saddle points of J^2. When the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$, R becomes the covariance matrix, then the above learning performs the analysis of the first k principal components (shortly k-PCA).

The above learning can have several variants. One variant is obtained by assuming that the vectors that span S are not a set of unit length orthogonal vectors, but a set of vectors that only satisfies $P = W(W^TW)^{-1}W^T = WW^T$. In this case, the problem of eq.(17) will become the unconstrained minimization

$$J^2 = \sum_{i=1}^{N} |\vec{x}_i - WW^T\vec{x}_i|^2. \qquad (18)$$

This is actually the special case of the LMSER proposed in [13] [17] for one layer linear network. From the studies of [13] [17], we know that the minimization of this J_2 will let W find the principal subspace spanned by the eigenvectors corresponding to the k largest eigenvalues of the correlation matrix R. When the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$, R becomes the covariance matrix, then the above learning performs the PSA problem studied in [11] [17] [21].

A more general variant is to let S spanned by a set of any independent vectors. In this case, the projector $P = W(W^TW)^{-1}W^T \neq WW^T$ and the problem of eq.(17) will become the minimization of

$$J^2 = \sum_{i=1}^{N} |\vec{x}_i - W(W^TW)^{-1}W^T\vec{x}_i|^2 \qquad (19)$$

From $\nabla_W J^2 = 0$, we can also show that the performance of the learning is the same as the one given by eq.(17).

Furthermore, let us consider the cases that $p(\vec{x})$ is specified by a linear manifold $S_M = \{\vec{y} : \vec{y} = \vec{x} + \vec{a}, \vec{x} \in S\}$, i.e., $M(\vec{x}) = S_M$. When $q = 2$, the error eq.(2) represents the perpendicular distance of $\vec{x}_i - \vec{a}$ to S, which is given by $|\vec{x}_i - \vec{a} - P(\vec{x}_i - \vec{a})|^2$. As a result, eq.(17) becomes

$$J^2 = \sum_{i=1}^{N} |(I - WW^T)(\vec{x}_i - \vec{a})|^2_{W^TW=I}. \qquad (20)$$

This minimization of J_2 with respect to both $W, \vec{a}$ will lead to PCA even when the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i \neq 0$. The solving of mean by $\vec{a}$ and the search of principal components by W are made in the same time.

Similarly, for the above linear manifold cases, eq.(18) and eq.(19) will respectively become

$$J^2 = \sum_{i=1}^{N} |\vec{x}_i - \vec{a} - WW^T(\vec{x}_i - \vec{a})|^2,$$

$$J^2 = \sum_{i=1}^{N} |\vec{x}_i - \vec{a} - W(W^TW)^{-1}W^T(\vec{x}_i - \vec{a})|^2 \qquad (21)$$

This minimization of these J_2 with respect to both $W, \vec{a}$ will lead to PSA even when we have the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i \neq 0$. The solving of mean by $\vec{a}$ and the principal subspace by W are made in the same time. Here, we do not need to first substruct the mean from each $\vec{x}_i$ and then start to learn the principal subspace, as it was made in [11] [17] [21].

Finally, we consider the cases that $p(\vec{x})$ is specified by the orthogonal complement subspace $\bar{S}$ of S. When $q = 2$, the error eq.(2) is

$$\min_{\vec{y}\in M(\vec{x},W)} |\vec{x}_i - \vec{y}|^2 = |WW^T\vec{x}|^2_{W^TW=I},$$

and eq.(17) becomes

$$J^2 = \sum_{i=1}^{N} |WW^T\vec{x}_i|^2_{W^TW=I}, \qquad (22)$$

Similar to the analysis of eq.(17), we will see that the minimization of eq.(22) will let W find the k eigenvectors corresponding to the k smallest eigenvalues of R. When the $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$ and thus R becomes the covariance matrix, the above learning performs the analysis of finding the first k minor components (shortly k-MCA).

Similarly, eq.(18) will become

$$J^2 = \sum_{i=1}^{N} |W(W^TW)^{-1}W^T\vec{x}_i|^2 = \sum_{i=1}^{N} \vec{x}\,^t W(W^TW)^{-1}W^T\vec{x}_i \qquad (23)$$

The minimization of J_2 will let W find the minor subspace spanned by the eigenvectors corresponding to the k smallest eigenvalues of R. When $\frac{1}{N}\sum_{i=1}^N \vec{x}_i = 0$ and R becomes the covariance matrix, the learnings perform the Minor Subspace Analysis (MSA).

Furthermore, when $p(\vec{x})$ is specified by a linear manifold $\bar{S}_M = \{\vec{y} : \vec{y} = \vec{x} + \vec{a}, \vec{x} \in \bar{S}\}$, eq.(21) and eq.(21) will become

$$J^2 = \sum_{i=1}^{N} |WW^T(\vec{x}_i - \vec{a})|^2_{W^TW=I},$$

$$J^2 = \sum_{i=1}^{N} |W(W^TW)^{-1}W^T(\vec{x}_i - \vec{a})|^2 \qquad (24)$$

This minimization of these J_2 with respect to both $W, \vec{a}$ will lead to k-MCA or MSA even when the mean $\frac{1}{N}\sum_{i=1}^N \vec{x}_i \neq 0$. The solving of mean by $\vec{a}$ and the search of minor components by W are made in the same time.

V. Unsupervised Learning: Multiple Models

We will consider the cases of model eq.(1) that $p(\vec{x})$ is specified by a number of points, lines, hyperplanes, subspaces, and linear manifolds, and show how the proposed theory perform various unsupervised learning tasks including *clustering, localized PCA localized MCA* and *localized PSA and MSA*.

A. Points: clustering

When $p(\vec{x})$ is specified by parametric points $\vec{a}_1, \cdots, \vec{a}_M$. The learning problem eq.(4) will become the minimization of

$$J^q_M = \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)|\vec{x}_i - \vec{a}_m|^q$$

$$+\beta\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)\ln\omega_m(\vec{x}_i), \quad s.t. \ \sum_{m=1}^{M}\omega_m(\vec{x}_i)=1. \qquad (25)$$

When $q=2$, eq.(5) will become

$$\omega_m^{(k)}(\vec{x}_i)=e^{-\beta^{-1}|\vec{x}_i-\vec{a}_m^{(k)}|^2}\Big/\sum_{m=1}^{M}e^{-\beta^{-1}|\vec{x}_i-\vec{a}_m^{(k)}|^2}. \qquad (26)$$

and the minimization of eq.(6) can be explicitly solved as

$$\vec{a}^{(k)}=\frac{1}{N}\sum_{i=1}^{N}\omega_m^{(k)}(\vec{x}_i)\vec{x}_i. \qquad (27)$$

Comparing the EM algorithm for clustering [20], we will find that eq.(26) is just the E-step and eq.(27) is just the M-step of the EM algorithm for clustering data points from M Guassian distributions with equal prior probabilities and given equal covariance $\Sigma_m=0.5\beta I$. As pointed out in [20], the widely used K-means clustering algorithm works also under this assumption, and the EM algorithm can even outperform the K-means algorithm. In other words, the learning eq.(4) will perform data clustering well.

B. Lines: local PCA

When $p(\vec{x})$ is specified by a set of lines that pass through points $\vec{a}_1,\cdots,\vec{a}_M$ respectively. We have M models specified as $M_m(\vec{x},\vec{w}_m)=\{\vec{x}:\vec{x}-\vec{a}_m=c_m\vec{w}_m,\ c_m\ \textit{is an arbitrary constant}\}$, for $m=1,\cdots,M$. For $q=2$, the learning problem eq.(4) will become the minimization of

$$J_M^q=\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)|(\vec{x}_i-\vec{a}_m)-\vec{w}_m\vec{w}_m^{\,t}(\vec{x}_i-\vec{a}_m)|^2$$
$$=\beta\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)\ln\omega_m(\vec{x}_i), \quad s.t. \ \sum_{m=1}^{M}\omega_m(\vec{x}_i)=1. \qquad (28)$$

with eq.(5) becoming

$$\omega_m^{(k)}(\vec{x}_i)=\frac{e^{-\beta^{-1}|(\vec{x}_i-\vec{a}_m^{(k)})-\vec{w}_m^{(k)}(\vec{w}_m^{(k)})^{\,t}(\vec{x}_i-\vec{a}_m^{(k)})|^2}}{\sum_{m=1}^{M}e^{-\beta^{-1}|(\vec{x}_i-\vec{a}_m^{(k)})-\vec{w}_m^{(k)}(\vec{w}_m^{(k)})^{\,t}(\vec{x}_i-\vec{a}_m^{(k)})|^2}}. \qquad (29)$$

and the minimization of eq.(6) being explicitly solved such that $\vec{a}_m^{(k)}$ is given by eq.(27) but using the above $\omega_m^{(k)}(\vec{x}_i)$ and $\vec{w}_m^{(k)}$ is the eigenvector of the matrix

$$\Sigma_m^{(k)}=\frac{1}{N}\sum_{i=1}^{N}\omega_m^{(k)}(\vec{x}_i)(\vec{x}_i-\vec{a}_m^{(k)})(\vec{x}_i-\vec{a}_m^{(k)})^T, m=1,\cdots,N. \qquad (30)$$

corresponding to the largest eigenvalue.

The result of the two-step iteration will finally let each of $\vec{a}_1,\cdots,\vec{a}_M$ converge to each center's location of M clusters of the data and each of $\vec{w}_1,\cdots,\vec{w}_M$ converge to the principal component of the corresponding clusters. Thus, we say that this learning performs *local PCA*. Recently, different implementations of local PCA have been suggested by [19] [6][8] and G.E.Hinton[2].

[2]Personal communication.

C. Hyperplanes: local MCA

Now, we consider the cases that $p(\vec{x})$ is specified by a set of hyperplanes $\vec{w}_m^{\,t}(\vec{x}-\vec{a}_m)=0$. When $q=2$, the learning problem eq.(4) will become the minimization of

$$J_M^q=\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)\frac{[(\vec{x}_i-\vec{a}_m^{(k)})^T\vec{w}_m^{(k)}]^2}{|\vec{w}_m^{(k)}|^2}$$
$$+\beta\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)\ln\omega_m(\vec{x}_i), \quad s.t. \ \sum_{m=1}^{M}\omega_m(\vec{x}_i)=1. \qquad (31)$$

with eq.(5) becoming

$$\omega_m^{(k)}(\vec{x}_i)=\frac{e^{-\beta^{-1}[(\vec{x}_i-\vec{a}_m^{(k)})^T\vec{w}_m^{(k)}]^2/|\vec{w}_m^{(k)}|^2}}{\sum_{m=1}^{M}e^{-\beta^{-1}[(\vec{x}_i-\vec{a}_m^{(k)})^T\vec{w}_m^{(k)}]^2/|\vec{w}_m^{(k)}|^2}}. \qquad (32)$$

and the minimization of eq.(6) being explicitly solved such that $\vec{a}_m^{(k)}$ is given by eq.(27) but using the above $\omega_m^{(k)}(\vec{x}_i)$, and $\vec{w}_m^{(k)}$ is the eigenvector corresponding to the smallest eigenvalue of the matrix $\Sigma_m^{(k)}$ given by eq.(30) but using the above $\omega_m^{(k)}(\vec{x}_i)$.

The result of the two-step iteration will finally let each of $\vec{a}_1,\cdots,\vec{a}_M$ converge to each center's location of M clusters and each of $\vec{w}_1,\cdots,\vec{w}_M$ converge to the minor component of the corresponding clusters. Thus, we say that this learning performs *local MCA*, which is an extension to MCA[14]. One application of this learning is to make piecewise fitting of curves and hypersurfaces[14].

D. Linear manifolds: local PSA and MSA

When $p(\vec{x})$ is specified by a set of linear manifolds. i.e., $M_m(\vec{x},W_m,\vec{a}_m)=\{\vec{y}:\vec{y}=\vec{x}+\vec{a},\vec{x}\in S_m\}, m=1,\cdots,M$ with S_m being a subspace specified by $W_m=[\vec{w}_{1m},\cdots,\vec{w}_{k_m m}]$. For $q=2$, the error eq.(2) becomes

$\varepsilon^q(\vec{x}_i,W_m,\vec{a}_m)=|\vec{x}_i-\vec{a}_m-W_mW_m^T(\vec{x}_i-\vec{a}_m)|^2$.

Putting it into eq.(4), the learning problem becomes the minimization of

$$J_M^q=\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)|(\vec{x}_i-\vec{a}_m)-\vec{W}_m\vec{W}_m^{\,t}(\vec{x}_i-\vec{a}_m)|^2$$
$$+\beta\sum_{m=1}^{M}\sum_{i=1}^{N}\omega_m(\vec{x}_i)\ln\omega_m(\vec{x}_i), \quad s.t. \ \sum_{m=1}^{M}\omega_m(\vec{x}_i)=1. \qquad (33)$$

with eq.(5) becoming

$$\omega_m^{(k)}(\vec{x}_i)=\frac{e^{-\beta^{-1}|(\vec{x}_i-\vec{a}_m^{(k)})-\vec{W}_m^{(k)}(\vec{W}_m^{(k)})^{\,t}(\vec{x}_i-\vec{a}_m^{(k)})|^2}}{\sum_{m=1}^{M}e^{-\beta^{-1}|(\vec{x}_i-\vec{a}_m^{(k)})-\vec{W}_m^{(k)}(\vec{W}_m^{(k)})^{\,t}(\vec{x}_i-\vec{a}_m^{(k)})|^2}}. \qquad (34)$$

and the minimization of eq.(6) being explicitly solved such that such that $\vec{a}_m^{(k)}$ is given by eq.(27) but using the above $\omega_m^{(k)}(\vec{x}_i)$ and $W^{(k)}=\Phi R$ with R being any $k_m\times k_m$ rotation matrix and the the column vectors of Φ consisting of the eigenvectors corresponding to the first k_m largest eigenvalues of $\Sigma_m^{(k)}$ given by eq.(30) but using the above $\omega_m^{(k)}(\vec{x}_i)$.

The result of the two-step iteration will finally let each of $\vec{a}_1,\cdots,\vec{a}_M$ converge to each center's location of M clusters and each W_m find the principal subspace of the corresponding clusters. Thus, we say that this learning performs *local PSA*, which is an extension to PSA [11] [17] [21].

Furthermore, considering the cases that $p(\vec{x})$ is specified by a set of linear manifolds $M_m(\vec{x},W_m,\vec{a}_m,)=\{\vec{y}:\vec{y}=\vec{x}+$

$\vec{a}, \vec{x} \in \bar{S_m}\}$, $m = 1, \cdots, M$ with $\bar{S_m}$ denoting the orthogonal complement subspace of the above S_m. For $q = 2$, the error eq.(2) becomes

$$\varepsilon^q(\vec{x}_i, W_m, \vec{a}_m) = |\vec{x}_i - \vec{a}_m - W_m W_m^T(\vec{x}_i - \vec{a}_m)|^2,$$

and the learning problem eq.(4) becomes the minimization of

$$J_M^q = \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)|W_m(W_m^T W_m)^{-1}W_m^T(\vec{x}_i - \vec{a}_m)|^2$$
$$= \beta \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)\ln \omega_m(\vec{x}_i), \ s.t. \ \sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1. \quad (35)$$

It can be implemented by the two iterative steps eq.(5) and eq.(6). Finally, each of $\vec{a}_1, \cdots, \vec{a}_M$ converge to each center's location of M clusters and $W_m = \Phi R$ with R being any $k_m \times k_m$ rotation matrix and the column vectors of Φ consisting of the eigenvectors corresponding to the first k_m largest eigenvalues of the covariance matrix of the corresponding clusters. In other words, W_m found the minor subspace of the corresponding clusters. Thus, we say that this learning performs *local MSA*, which is an extension to MSA[21].

VI. Self-Organizing Maps

Another type of widely studied unsupervised learning is topologically preserved self-organizing map[7][?]. In the following, we will show that the proposed model eq.(4) can also be extended to generating this map.

Let us to consider the case for clustering given by eq.(25). An one dimensional self-organizing map can be generated by modifying eq.(25) as follows

$$J_M^q = \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)|\vec{x}_i - \vec{a}_m|^q$$
$$+\beta \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)\ln \omega_m(\vec{x}_i) + \gamma \sum_{m=1}^{M-1} |\vec{a}_{m+1} - \vec{a}_m|^q \quad (36)$$

with s.t.$\sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1$, where $\gamma > 0$ is a given parameter. $\vec{a}_1, \cdots, \vec{a}_M$ are ordered sequentially in a line structure, and the third term e is to force the parameters of two topologically nearby neurons are also close to each other.

Furthermore if we reorder $\vec{a}_1, \cdots, \vec{a}_M$ in to an array $\vec{a}_{ij}, i = 1, \cdots, M_1, i = 1, \cdots, M_2$, and also modify eq.(36) into

$$J_M^q = \sum_{m=1}^{M_1}\sum_{r=1}^{M_2}\sum_{i=1}^{N} \omega_{mr}(\vec{x}_i)|\vec{x}_i - \vec{a}_{mr}|^q$$
$$+\beta \sum_{m=1}^{M_1}\sum_{r=1}^{M_2}\sum_{i=1}^{N} \omega_{mr}(\vec{x}_i)\ln \omega_{mr}(\vec{x}_i)+$$
$$\gamma \sum_{m=1}^{M_1-1}\sum_{r=1}^{M_2-1} [|\vec{a}_{m+1,r} - \vec{a}_{m,r}|^q + |\vec{a}_{m,r+1} - \vec{a}_{m,r}|^q] \quad (37)$$

with $s.t.$ $\sum_{m=1}^{M_1}\sum_{r=1}^{M_2} \omega_{mr}(\vec{x}_i) = 1$. We can also get a two dimensional self-organizing map.

Similarly, the idea can also be applied to other cases. For example, we can modify eq.(28) into

$$J_M^q = \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)|(\vec{x}_i - \vec{a}_m) - \vec{w}_m\vec{w}_m^t(\vec{x}_i - \vec{a}_m)|^2$$
$$+\beta \sum_{m=1}^{M}\sum_{i=1}^{N} \omega_m(\vec{x}_i)\ln \omega_m(\vec{x}_i) + \gamma[\sum_{m=1}^{M-1} |\vec{a}_{m+1} - \vec{a}_m|^2$$
$$+ \sum_{m=1}^{M-1} \frac{(\vec{w}_{m+1}^T\vec{a}_m)^2}{|\vec{w}_{m+1}|^2|\vec{w}_m|^2}], \ s.t. \ \sum_{m=1}^{M} \omega_m(\vec{x}_i) = 1. \quad (38)$$

to get an one dimensional self-organizing map, where pairs $(\vec{a}_1, \vec{w}_1), \cdots, (\vec{a}_M, \vec{w}_M)$ are ordered sequentially in a line structure. In the same way as eq.(36), we can also turn eq.(28) into a model for a two dimensional self-organizing map.

References

[1] S. Becker. *Int. J. of Neural Systems*, vol. 2, 1991, 17-33.

[2] A.P.Dempster, N.M. Laird and D.B.Rubin, *J. of Royal Statistical Society, B39*, 1977 pp1-38.

[3] R.J.Hathaway, *Statistics & Probability Letters 4*, 1986, pp53-56.

[4] R.A.Jacobs, M.I.Jordan, S.J. Nowlan,and G.E.Hinton, *Neural Computation, 3*, 1991, pp 79-87.

[5] M.I.Jordan and R.A.Jacobs, *Advances in Neural Information Processing System 4*, eds., J.E.Moody, S.Hanson and R.P.Lippmann, San Mateo: Morgan Kaufmann Pub., 1992, pp 985-992.

[6] M.I.Jordan and R.A.Jacobs, Hierarchies mixtures of experts and the EM algorithm, MIT Computational Cognitive Science, Tech. Rep. 9203, MIT, 1992.

[7] T. Kohonen. *Self-Organization and Associative Memory*, 3rd ed. Springer-Verlag, 1989.

[8] T.K.Leen and N.Kambhatla, Fast non-linear dimension reduction, Tech. Rep. No. CS/E 93-011, Oregon Grad. Inst. of Sci. & Tech., Beaverton, OR, 1993.

[9] R.M. Neal and G.E. Hinton (1993), A new view of the EM algorithm that justifies incremental and other variants, submitted to Biometrika. 1993.

[10] E. Oja. *J. of Math. Biology*, vol. 16, 1982, 267-273.

[11] E. Oja. *Int. J. of Neural Systems*, vol. 1, 1989, 61-68.

[12] E. Oja. *Neural Networks*, vol. 5, 1992, 927-936.

[13] L. Xu. In *Proc. IJCNN'91*, Singapore, November 1991, pp. 2362-2367 (part I) and pp. 2368-2373 (part II).

[14] L. Xu and E. Oja, *Neural Networks*, vol. 5, 1992, 441-457.

[15] L. Xu and A. Yuille, In *Proc. IJCNN'92*, Baltimore, Maryland, June 1992, I-812-817, also in S.J. Hanson, J.D. Cowan, and C.L. Giles (Eds.), *Advances in NIPS 5*. Morgan Kaufmann, San Mateo, CA, 1993, 467-474.

[16] L. Xu , S.Klasa and A.L. Yuille, *Int. J. of Neural Systems*, Vol.3, 1992, 253-290.

[17] L. Xu, *Neural Networks*, vol. 6, 1993, 627-648.

[18] L.Xu, M. I. Jordan and G.E. Hinton, A modified gating network for the mixtures of experts architecture, submitted to WCNN'94, San Diego, 1993.

[19] L.Xu and M.I.Jordan, Theoretical and Experimental Studies of The EM Algorithm for Unsupervised Learning Based on Finite Gaussian Mixtures, MIT Computational Cognitive Science, Tech. Rep. 9301, MIT, Cambridge, MA, 1993.

[20] L.Xu and M.I.Jordan, Proc. WCNN'93, Portland, OR, Vol. II, 1993, 431-434.

[21] L.Xu, Theories for unsupervised learning: pca and its nonlinear extensions, will be presented also in this IEEE ICNN'94 (invited paper), 1994.

[22] A.L. Yuille and J.J. Kosowsky, Statistical physics algorithms that converge, Harvard Robotics Lab. Tech. Rep. 92-7. 1992.