BYY Harmony Learning, Independent State Space, and Generalized APT Financial Analyses

Lei Xu, *Fellow, IEEE*

***Abstract*—First, the relationship between factor analysis (FA) and the well-known arbitrage pricing theory (APT) for financial market has been discussed comparatively, with a number of to-be-improved problems listed. An overview has been made from a unified perspective on the related studies in the literatures of statistics, control theory, signal processing, and neural networks. Second, we introduce the fundamentals of the Bayesian Ying Yang (BYY) system and the harmony learning principle which has been systematically developed in past several years as a unified statistical framework for parameter learning, regularization and model selection, in both nontemporal and temporal stochastic environments. We further show that a specific case of the framework, called BYY independent state space (ISS) system, provides a general guide for systematically tackling various FA related learning tasks and the above to-be-improved problems for the APT analyses. Third, on various specific cases of the BYY ISS system in three typical architectures, adaptive algorithms, regularization methods and model selection criteria are provided for either or both of parameter learning with automated model selection and parameter learning followed by model selection. In the B-architectures, new results are provided for Gaussian and non-Gaussian FA, binary FA, independent Hidden Markov Model (HMM) and Temporal FA, as well as other extensions, which are then applied to statistical APT analyses for solving the above to-be-improved problems. In the F-architectures, adaptive algorithms are given for several extensions of independent component analysis (ICA), including competitive ICA, Gaussian and non-Gaussian temporal ICA. Moreover, the advantages of the B-architectures and the F-architectures are traded off in the BI-architectures, not only with new strength to the existing least mean square error reconstruction (LMSER) learning, but also with various LMSER extensions, including the so-called principal ICA and its temporal extension. The final part of this paper introduces some other financial applications that base on the underlying independent factors via the APT analyses, including prediction of macroeconomic indexes, portfolio management by adaptively maximizing an adjusted Shape ratio, and a macroeconomics modulated independent state-space model for financial market modeling.**

***Index Terms*—Arbitrage pricing, BYY system, data-smoothing, factor analysis, financial modeling, finite sample size, harmony learning, hidden Markov model, ICA, independence, LMSER learning, normalization, portfolio, regularization, source separation, state space.**

Manuscript received September 1, 2000; revised March 15, 2001 and March 28, 2001. This work was supported by the Research Grant Council of the Hong Kong SAR (Project CUHK 4169/00E).

The author is with Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong, P.R. China.

Publisher Item Identifier S 1045-9227(01)05013-5.

I. Factor Analysis, APT Theory, and Related Literatures

IT has been a well-known philosophy that a complicated observation is regarded as generated from a number of hidden and simpler factors via certain transformation or mixing system such that the observation can be understood by recovering the factors via an inverse transformation or demixing system. Many efforts have also been made on developing mathematical theories that implements this philosophy to solve various practical problems in a number of scientific and engineering fields.

The earliest effort can be traced back to the beginning of the 20th century by Spearman [54], and had been followed by various studies in the literature of statistics, which use the following linear model:

$$x_t = Ay_t + e_t, \quad E(e_t) = 0, \quad e_t \text{ is independent from } y_t \tag{1}$$

where and throughout this paper, the notations $E(u) = Eu = E[u]$ denotes the expectation of random variable u. The model (1) has been applied to various explanatory modeling tasks in sciences, especially behavioral and social sciences. In this simple model, a random sample $x_t = [x_t^{(1)}, \cdots, x_t^{(d)}]^T$ of observation is generated via a linear mapping matrix A from k hidden factors in the form $y_t = [y_t^{(1)}, \cdots, y_t^{(k)}]^T$, disturbed by a noise e_t as given in (1). Usually, samples of e_t are independently and identically distributed (i.i.d.) from a same probability density function (pdf) p_e. The general ambition is to determine A and the statistics of y_t and e_t from a series of samples $\mathbf{x} = \{x_t\}_{t=1}^T$. Obviously, the problem is not well defined because there are an infinite number of solutions. To reduce the indeterminacy, we consider that samples of x_t are i.i.d. and correspondingly samples of y_t are also i.i.d. from a pdf p_y. Hence, x_t can also be modeled by

$$p(x_t) = \int p_e(x_t - Ay_t)p(y_t)\, dy_t. \tag{2}$$

Usually, the above treatment is still not enough to make the problem sensible, and extra constraints must be adequately imposed. Specifically, different types of constraints will lead to different statistical approaches that implement (1), which are briefly summarized as follows:

- **Linear Regression:** When it is also possible to know the corresponding series $\mathbf{y} = \{y_t\}_{t=1}^T$, the task becomes the typical linear regression problem. When the pdf form of p_e is known, A can be determined by using the classical maximum likelihood (ML) method on $p_e(x_t - Ay_t)$, which can be further simplified into the conventional least square

method or the weighted least square method when p_e is Gaussian.

- **Inverse Problem:** When both A and p_e are known, the task becomes the typical inverse problem that maps each x_t into a corresponding estimate $\hat{y}_t$. One typical technique is again the ML method

$$\begin{aligned}\hat{y}_t &= \arg\max_{y_t} \ln p_e(x_t - Ay_t), \text{ e.g.,} \\ &\quad \text{for } p_e = G(e|0,\Sigma), \quad y_t = Wx_t, \\ W &= (A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}, \\ \Sigma &= \frac{1}{N}\sum_{t=1}^{N}(x_t - Ay_t)(x_t - Ay_t)^T \end{aligned} \tag{3}$$

where and throughout this paper, $G(u|m,\Sigma)$ denotes a Gaussian density with mean m and covariance Σ, and $\hat{r} = \arg\max_r f(r)$ denotes the value of r that makes $f(r)$ be maximum. Particularly, we have $W = (A^TA)^{-1}A^T$ when $\Sigma = \sigma^2 I$, which is usually called the least square inverse. When $\Sigma \neq \sigma^2 I$, W, Σ can be estimated iteratively via the so called reweighted least square inverse.

- **Factor Analysis:** Instead of imposing that either A or y_t is known, an alternative way is to impose certain structures on y_t and e_t such that the indeterminacy of (1) could be reduced to a level that (1) becomes meaningful to certain applications. One typical example is well known as factor analysis [40], [50]. Formulated by Anderson and Rubin in 1956 [4], both e_t, y_t come from Gaussian with $E(e_t) = 0$, $E(y_t) = 0$. Specifically, e_t is uncorrelated among its components with a diagonal covariance matrix Σ_e. Moreover, e_t is uncorrelated to y_t that is itself uncorrelated among its components. Furthermore, it is usually assumed that $E(y_t y_t^T) = I$ since the uncorrelated components remain uncorrelated after any scaling transform

$$Ay_t = A'y'_t, \quad A' = AD^{-1}, \quad y'_t = Dy_t, \quad D \text{ is diagonal.} \tag{4}$$

In such a formulation, the pdf-based equation (2) becomes

$$\begin{aligned}\Sigma_x = AA^T + \Sigma_e, &\Sigma_x, \Sigma_e \text{ are covariance matrices} \\ &\text{of } x, e \text{ respectively.}\end{aligned} \tag{5}$$

This matrix equation may still have many solutions due to the following two types of indeterminacy:

1) *Rotation indeterminacy* For any rotation matrix $z_t = \psi y_t$, $\psi\psi^T = I$, obviously we have

$$E(y_t y_t^T) = E(z_t z_t^T) = I, \quad \text{or} \quad A\psi\psi^T A^T = AA^T. \tag{6}$$

2) *Additive indeterminacy* The additive symmetry of the two items in the left side of (5) makes the indeterminacy of decomposing the diagonals of Σ_x into the diagonals of Σ_e and of AA^T. It is also called the communality estimation problem [40].

Two heuristic ways are usually used to remove these types of indeterminacy. One is simply set $\psi = I$. That is, in help of the singular value decomposition $A = \phi\Lambda\psi^T$, $\phi\phi^T = I$, $\psi\psi^T = I$ with a diagonal Λ, it follows from (6) that $A' = \phi\Lambda$ is a solution with $A'A'^T = AA^T$. Also, the additive indeterminacy is removed when the ML learning is made on $p(x) = \int G(x|Ay, \Sigma_e)G(y|0,I)\,dy$. Furthermore, in the case $\Sigma_e = \sigma_e^2 I$, we even get an analytical solution that ϕ consists of the first k component eigen-vectors of the sample covariance S_x, a diagonal Λ consists of the corresponding eigen-values, and σ_e^2 is an average of the last $d-k$ eigen-values [67]. The other heuristic way is to select a specific rotation instead of imposing $\psi = I$. Typical examples include Quartimax and Varimax, which have been used in the literature of statistics [40].

- **Principle Component Analysis:** Instead of getting $\hat{y}_t$ based on knowing A and the covariance matrices of y_t, e_t, an alternative is to get $\hat{y}_t = Wx_t$ under the orthogonal constraint $W^TW = I$ such that $\hat{y}_t$ becomes uncorrelated in components and the variance of each component is maximized. The solution is analytically given by $W = \phi$, where ϕ again consists of the first k components of S_x. This is well known as principle component analysis (PCA), which can be backtracked to as early as in 1936 by Hotelling [31], and has been also widely studied in the literatures of statistics, pattern recognition and neural networks [43]. Putting the above solution $A = \phi\Lambda$ at $\Sigma_e = \sigma_e^2 I$ into (3), we have $W = (A^TA)^{-1}A^T = \Lambda^{-1}\phi^T$. That is, the least square inverse of this specific factor analysis is actually equivalent to PCA up to a scale difference by Λ^{-1}.

Interestingly, the same model (1) has also been approached from the perspective of finance theory. In the literature of financial market modeling, the well-known arbitrage pricing theory (APT) is proposed by Ross in 1976 [47], [49]. According to APT, the return on security can be broken down into an expected return and an unexpected or surprise component, usually called news. This news can be further classified into two types. One is general news that affects all stocks, e.g., an unexpected announcement of an interest rate change by the government. The other is specific news that affects particular stocks. The APT theory believes that these general news will affect the rate of returns on all stocks by different degrees of sensitivity.

To be more specific, the APT infers that an individual asset is associated with multiple risky factors $\{f^{(i)}\}$ as follows:

$$E(r^{(j)}) = r^f + \sum_{i,j} a_{i,j}(E(f^{(i)}) - r^f) \tag{7}$$

where

$E(r^{(j)})$	expected return on a risky asset j;
r^f	risk-free return;
$a_{i,j}$	is the sensitivity of asset j to the risky factor i.

Considering the instantaneous form, (7) is rewritten into the following form with a residual return $e^{(j)}$, $E(e_t^{(j)}) = 0$ to each asset j:

$$\begin{aligned} r^{(j)} - r_t^f &= \sum_{i,j} a_{i,j}(f_t^{(i)} - r^f) + e_t^{(j)}, \text{ or} \\ x_t^{(j)} &= \sum_{i,j} a_{i,j} y_t^{(i)} + e_t^{(j)} \\ x_t^{(j)} &= r^{(j)} - r_t^f, \quad y_t^{(i)} = f_t^{(i)} - r^f. \end{aligned} \tag{8}$$

Obviously, its matrix form is exactly (1).

Since its inception in the mid-1970s, the APT has attracted a considerable interest as a tool for interpreting investment results and controlling portfolio risk [25], [1], [18]–[20], [52]. To implement the APT, the key is to determine what are used as the factors. Three approaches are usually applied for the purpose:

- **Time series approach** that directly uses a historic time series of a set of macroeconomic or fundamental indexes as the series of factors $\mathbf{y} = \{y_t\}_{t=1}^T$. These factors are usually known as *fundamental factors*, including GDP, inflation, interest rate, oil price, ... etc. In this case, a typical time series regression is made to estimate A, which it is a typical linear regression problem.
- **Cross-sectional approach** that begins with estimates of elements of A which are usually called the attributes or the securities' sensitivities to y_t, known as empirical factors. The attributes are obtained via observing the correlation between x_t, y_t. The task is to estimate y_t upon x_t, which can be made again by a linear regression on (1) at the current time t. Actually, it is equivalent to set up an inverse mapping (3).
- **Factor-analytic approach** that uses the factor analysis approach to get both the unknown A and the unknown factors y_t estimated from the observed time series $\mathbf{x} = \{x_t\}_{t=1}^T$.

Despite its attractive features, the APT has not been widely applied by the investment community. The reason lies largely with the APTs most significant drawback: the lack of specificity regarding the factors that systematically affect security returns. In the uses of the above first two approaches, either fundamental factors or attributes are chosen heuristically and even quite arbitrarily, based on preknowledge or beliefs. In the use of factor-analytic approaches, there is no need on external heuristics. Thus, they become more appealing in the general cases where we have only the observed price movements x_t [25], [1], [18]–[20]. Unfortunately, certain empirical tests [25], [1] showed that factor analysis does not specify what economic variables that the factors represent.

There remains a plenty room for improving the implementation of APT. The failures of the factor-analytic approach may be due to both inappropriate tools for handling the intrinsic indeterminacy in (1) and the inadequacy of (1) for modeling a time series $\mathbf{x} = \{x_t\}_{t=1}^T$, which are summarized as follows:

- *Problem (a) Three types of indeterminacy* Though the scaling indeterminacy (4) can be acceptably removed by considering uncorrelated factors with unit variance, both the rotation indeterminacy and additive indeterminacy make the factor-analytic approaches not able to specify an appropriate solution. The above discussed ways of imposing $\psi = I$ or Quartimax and Varimax are too arbitrary to provide a good solution.
- *Problem (b) How to determine the number k of factors* The selection of a correct number of factors are essential to the performances of using the APT model [20]. But it is usually set heuristically.
- *Problem (c) Ignorance of temporal relation* The above discussed implementation of (1) implies assuming that each x_t for different t is i.i.d. and correspondingly each y_t for different t is also i.i.d. However, in practice there is a temporal or serial relation among the samples of $\mathbf{x} = \{x_t\}_{t=1}^T$, which should not be ignored.
- *Problem (d) Non-Gaussian noise* The existing studies on (1) consider the cases that the unknown e_t in (1) is Gaussian, which is not suitable for the cases that e_t is non-Gaussian.
- *Problem (e) Nonadditive model* Equation (1) considers two independent terms additively, which is not applicable to the case that it is impossible to decompose x_t into two independent additive terms.
- *Problem (f) Nonlinear model* The linear model (1) is not adequate for the cases that x_t, y_t have a nonlinear regression relation.

It should be noticed that the above problems are essential not only specifically to the performance of the APT implemented by statistical factor analysis but also generally to the success of the factor model (1) on various practical applications. In addition, appropriate solutions of the above problems (a)–(f) also provide improvements and extensions on the performance of the APT implemented by the cross-sectional approach. The paper is motivated to tackle these problems systematically. Actually, some problems have been partially touched already, scattered in different literatures. To provide a background for a better understanding on the work of this paper, we make a comparative overview on the existing major advances.

- **Independent Component Analysis (ICA) and Blind Source Separation (BSS):** Started from Jutten and Herault in 1988 [35], a simplified model of (1) is considered by setting $d = k$ and $e_t = 0$, i.e., $x_t = Ay_t$ with A being a unknown invertible matrix. They assume that the components of y are independent and non-Gaussian or with at most only one of them being Gaussian. This assumption removes out the rotation indeterminacy (6) because components can not keep independent after a rotation transform. Thus, $\hat{y}_t = Wx_t$ can recover y_t up to the scaling indeterminacy (4) if W makes nonGaussian $\hat{y}_t$ become component-wise independent [59], [17]. For this nature, it is called ICA, named in contrast to PCA. The rotation indeterminacy is removed by extracting the higher order information from data, instead of imposing extra heuristics while still based on the statistics up to second order only in the above mentioned Quartimax or Varimax [40]. Moreover, when each component of y_t is interpreted as a sample of a time series at the moment t, the fact that the ICA solution $\hat{y}_t = Wx_t$ recovers y_t up to the scaling indeterminacy (4) means that the waveform of each component series can be recovered. Thus, this recovery $\hat{y}_t = Wx_t$ is also said to perform BSS that blindly separates the mixed signal $x_t = Ay_t$.

Advances on ICA can be roughly summarized into several stages. First, several learning algorithms for estimating W have been proposed from different perspectives [35], [28], [17], [8], [3], [27]. Usually, these algorithms work well on the cases that the components of y are either all sub-Gaussians or all super-Gaussians because a prefixed pdf form is used as each $p(y_t^{(j)})$,

either heuristically (e.g., the sigmoid used in [8]) or based on kurtosis estimation or density expansion [17], [3]. At the second stage, it is realized [72] that the pdf form for each $p(y_t^{(j)})$ should also be learned simultaneously during learning on W such that whether a component is super-Gaussian or sub-Gaussian can be automatically detected in order to work on any combination of super-Gaussian or sub-Gaussian components of y. This idea has been implemented by learning $p(y_t^{(j)})$, during learning on W, via estimating the parameters θ_y of the following parametric model:

$$p(y_t|\theta_y) = \prod_{j=1}^{k} p(y_t^{(j)}|\theta_y^{(j)}). \tag{9}$$

An early effort of this type is called the learning parametric mixture based ICA [69], [72], [79], where a finite mixture is used as $p(y_t^{(j)}|\theta_y^{(j)})$, with $\theta_y^{(j)}$ updated by an EM-like algorithm during updating W. Alternatively, efforts [44], [16] have also been made on estimating the kurtosis of $p(y_t^{(j)})$. At the third stage, extensions have been made toward various general cases, e.g., (a) the dimension of x_t is larger than k instead of that A is invertible [70], [69], [68], (b) some specific nonlinear system $x_t = g(y_t)$ instead of the linear model (1) [57], [68]. A more detailed review on the advances of ICA is referred to [63], [69], [76], and [77]. Some interesting studies have also been made on using an ICA algorithm for exacting structures from stock returns by [5].

- **Non-Gaussian Factor Analysis:** It is not realistic to assume zero noise e_t. Thus, it is more reasonable to consider (1) with the independence assumption (9) in place of the previous assumption that components are decorrelated. For clarity, we refer this type of factor analysis (FA) as *Non-Gaussian FA* or *independent FA* to avoid being confused with the above discussed FA, which should now be more precisely referred as *Gaussian FA or de-correlating FA*. Similar to ICA, when each $p(y^{(j)})$ is non-Gaussian or at most only one of them is Gaussian [68], the rotation indeterminacy (6) can be removed. However, not as in Gaussian FA, it is not an easy task to estimate A as well as the parameters of $p(y_t^{(j)})$ and p_e by using the ML method on $p(x)$ in (2) because a computational difficulty will be encountered for implementing the integral (2), especially when y_t is a real vector. In [68], the integral is handled by a Monte-Carlo sampling method, based on which a general adaptive algorithm is proposed to make the ML learning. Also, as a solution of the *problem (b)*, a criterion is obtained from the so called Bayesian Ying-Yang (BYY) learning for detecting k. On the data with binary signals y_t, experiments have shown that the algorithm and criterion work well. Also in [68], an alternative way to get rid of the integral (2) is made via a mean-field type approximation. While in the literature of ICA studies, efforts have also been made on getting W that makes $\hat{y}_t = Wx_t$ become component-wise independent by taking x_t from (1) with $e_t \neq 0$ in consideration, in help of some heuristics or certain structures [16], [27]. These studies are usually called noisy ICA, which are closely related to non-Gaussian factor analysis.

Even further, as an effort toward to the *problem (d)*, the case that e_t in (1) is not Gaussian has been considered in [68] by using Gaussian mixture to model the pdf $p_e(x_t - Ay_t)$. In [68], the linear model (1) has been also extended to a general nonlinear factor model

$$\begin{aligned} x_t &= g(y_t, \theta_g) + e_t, \\ E(e_t) &= 0, e_t \text{ is independent from } y_t \end{aligned} \tag{10}$$

as an effort toward to the above *problem (f)*. Furthermore, in the literature of neural networks, efforts have also been made on modeling binary x_t (e.g., representing a binary image) by interpreting it as generated from binary hidden factor y_t with mutually independent bits. Typical examples include multiple cause models [51], [21] and Helmholtz machine [22], [30]. These studies can actually be regarded as efforts toward to the *problem (e)*. In this case, the regression between binary x_t and y_t is nonlinear and thus acts as $g(y_t, \theta_g)$ in (10), but $e_t = x_t - E(x_t|y_t)$ becomes not independent from y_t. That is, it can not be decomposed into two independent additive terms and described by (2). Actually, it is a specific case of the following general factor model:

$$p(x_t|\theta) = \int p(x_t|y_t)p(y_t)\, dy_t. \tag{11}$$

By regarding $g(y_t, \theta_g) = \int x_t p(x_t|y_t, \theta_g)\, dx_t$ and $e_t = x_t - g(y_t, \theta_g)$, we observe that (11) includes not only (10) but also those cases that e_t is not independent from y_t, such as the above one.

- **Hidden Markov Model (HMM):** As a popular topic in the literature of speech processing since the early 1970s [45], HMM is a classical example of the early efforts toward solutions to both the *problem (c)* and *problem (e)*, and has been proved to be one of most effective tools for modeling temporal relation with wide applications. In a classical HMM, x_t is discrete and described by (11) with $p(x_t|y_t)$ being a $d \times k$ transfer probability matrix, which is equivalent to a nonlinear regression $g(y_t, \theta_g)$ and $e_t = x_t - g(y_t, \theta_g)$ is not independent from each discrete value of y_t. The temporal relation is modeled by introducing the first-order Markov transfer probability $p(y_t = j|y_{t-1} = i)$ that turns $p(y_{t-1})$ into $p(y_t)$, which is a special case of the following general form of the order one Markov model:

$$p(y_t) = \int p(y_t|y_{t-1})p(y_{t-1})\, dy_{t-1}. \tag{12}$$

  In the HMM studies, $p(x_t|y_t)$ and $p(y_t|y_{t-1})$ are unknown and can be solved from observations by the Baum algorithm for the ML learning [45].
- **State-Space Model and Kalman Filter:** Another early effort toward solutions to the *problem (c)* is the classic state-space model:

$$\begin{aligned} y_t = By_{t-1} + \varepsilon_t, \quad x_t = Ay_t + e_t, \\ \varepsilon_t \text{ is independent from both } e_t \text{ and } y_t \end{aligned} \tag{13}$$

  which is studied in the literatures of control theory since the early 1960s [36]. The model (13) can be regarded as an

extension of (1), with the first equation added in for modeling temporal relation. The added equation represents the factor series $\mathbf{y} = \{y_t\}_{t=1}^T$ in a multichannel autoregressive process, driven by an i.i.d. noise series $\{\varepsilon_t\}_{t=1}^T$ that are independent of both y_{t-1} and e_t. Generally, the indeterminacy of (13) is more serious than the original model (1) because of the new unknowns of B and the statistical property of ε_t. Thus, further constraints are obviously needed.

The most well-known successful use of (13) is the Kalman filter that adaptively estimates $\hat{y}_t$ upon observing x_t under the condition that both A, B are known and both ε_t, e_t are Gaussian with known covariance matrices. The Kalman filter extends the task of the optimal linear mapping $x_t \to y_t$ by taking temporal relation in consideration. In the past three decades, Kalman filter has been widely used in the literatures of control theory and signal processing [13]. In recent years, efforts have been made on using Kalman filter for discovering the true pricing in financial series [10], [41], [42]. However, inherited from Kalman filter, these studies either fix the matrices A, B by some simple constants (e.g., $A = 1, B = 1$ in [41], [42]) or model them by a specific structure such that the unknowns are estimated also via Kalman filter [10], [34].

In [63], [76], and [77], as a new effort to the *problem (c)*, we implant the basic assumption (9) into the state-space model (13) and impose the independence assumption on the components of y_t, i.e.,

$$p(\mathbf{y}) = \prod_{j=1}^{k} p(\mathbf{y}^{(j)}), \quad p(y_t|\mathbf{y}_{t-1}) = \prod_{j=1}^{k} p(y_t^{(j)}|\mathbf{y}_{t-1}^{(j)})$$
$$\mathbf{y}_{t-1} = [y_{t-1}, \cdots, y_1]^T \tag{14}$$

where $\mathbf{y}^{(j)}$, $\mathbf{y}_{t-1}^{(j)}$ denotes the jth row of $\mathbf{y}$, $\mathbf{y}_{t-1}$, respectively, and $\mathbf{y}_{-1}$ is an empty set. On (13), it becomes the following condition:

$$B \text{ is diagonal and } \varepsilon_t \text{ is mutually independent in components} \tag{15}$$

which is called temporal factor analysis (TFA) model. Specifically, it is called either Gaussian TFA when ε_t is Gaussian or non-Gaussian TFA when ε_t is non-Gaussian. It has been shown in [63] that the use of temporal relation in (13) can also remove out the previous rotation indeterminacy. Moreover, for Gaussian TFA, an adaptive algorithm is proposed to learn A, B as well as the covariance of e_t via the ML estimation on $p(x_t)$ by (2) under the constraint of (12). Also, a criterion is obtained from temporal BYY learning for deciding k as an effort toward to the *problem (b)*. Particularly, in the special case $e_t = 0$, we also get an extension of ICA, named temporal ICA, that takes temporal relation in modeling, with an adaptive algorithm provided. Furthermore, when y_t is a binary vector, variants of HMM have been obtained to consider higher order temporal relation in implementation of binary non-Gaussian TFA and binary BSS with noise. Experiments have demonstrated that these temporal algorithms outperform their nontemporal counterparts significantly.

The rest of paper consists of two parts. The first part is given in Section II in which we describe the fundamentals of the BYY harmony learning that is proposed as a unified statistical learning theory first in 1995 [73], [80] and then systematically developed in the past years [62]–[64], [66], [68], [76], [77]. Specifically, the best harmony learning principle is presented in Section II-A for both parameter learning and model selection, with discussions on its relation to the ML learning and regularization. In Sections II-B–II-D, the BYY system and three typical architectures are introduced. Also, the key issues of the harmony learning on the BYY system are described in two types of implementations, namely, parameter learning with automated model selection and parameter learning followed by model selection. In Section II-E, the BYY harmony learning is further extended to stochastic environment, especially to the temporal BYY learning. In Section II-F we present a general adaptive learning procedure that applies to all the three typical BYY system architectures in typical specific structures.

The second part is given in Section III in which we introduce the detailed forms of the harmony learning on three typical architectures of the BYY independent state space system, and then apply them to statistical APT financial analyses with extensions. The backward architecture is introduced in Section III-A, with adaptive algorithms, regularization techniques and model selection criteria on various cases of the independent state space model (13) and their extensions, which include non-Gaussian FA, Gaussian and non-Gaussian TFA, binary FA, and independent HMM as well as their extensions to non-Gaussian observation noise and nonlinear models. All of them can not only be used for corresponding unsupervised learning tasks, but also be used as tools for implementing generalized APT financial analyses, with the previously listed *problems (a)–(f)* in consideration. In Section III-B, we first introduce the forward architecture, with adaptive algorithms on Gaussian and nonGaussian temporal ICA, as well as a competitive ICA for data of multiple modes. Then, we introduce the bidirectional architecture that trades off the advantages of a backward architecture and a forward architecture, with not only new strength to the existing LMSER learning [75], [82] but also to various LMSER extensions, including a type of principal ICA and its temporal extensions. Moreover, in Section III-C, the APT analyses is used for return prediction, macroeconomic modeling and particularly portfolio management by adaptively maximizing a modified Shape ratio in help of hidden factors. In Section III-D, as an alternative to APT, a macroeconomics modulated independent state-space model is proposed to for capital market modeling by taking macro-economy indices in consideration. Finally, we conclude in Section IV after providing demonstrative experiments in Section III-E.

II. Bayesian Ying Yang System and Harmony Learning

A. Best Harmony Learning Principle

In general, specifying a density $q(u)$ involves three issues. The first issue is a given structure. A typical example is given in Table I with y as u, which is a product structure that consists of components. Each component is either (a) a basic component, in a sense that its pdf form is given and there remains only a set of unknown parameters, or (b) a summation structure that itself consists of a number of components which are organized in a weighted sum. The task of *structure design* is to specify

the pdf forms of basic components and the structure that these basic components are organized. The second issue is the set $\mathbf{k} = \{k, \{\kappa_{j,r}\}\}$ that describes the scale of a given structure. The task of specifying the scale is called *model selection* in a sense that a collection of different scales corresponds a collection of specific models that share a same configuration but in different scales, and thus selecting a specific scale is equivalent to selecting a model. The *third issue* is a collection θ of unknown parameters. The task of specifying θ is called *parameter learning*.

In a conventional sense, learning is a process that specifies a density $q(u)$ from a given data set $\mathbf{u} = \{u_t\}_{t=1}^N$, via specifying $\theta, \mathbf{k}$ under a given structure design. Without extra *a priori* constraints, learning from $\mathbf{u}$ is equivalent to learn from its empirical density [24]:

$$p_0(u) = \frac{1}{N}\sum_{t=1}^{N}\delta(u-u_t),$$
$$\delta_u(u) = \begin{cases} \lim_{h\to 0} 1/h^k, & u = 0 \\ 0, & u \neq 0. \end{cases} \tag{16}$$

Instead of $p_0(u)$, from $\mathbf{u}$ we can also use a kernel density estimate, where each sample is smoothed under the control of an extra *a priori* smoothing parameter $h >$, as will later to be introduced in (25) and (33).

In a broad sense, we consider learning not only in this case but also in the cases that there are two $p(u)$, $q(u)$ in known structures but each of them having some unknown parts, e.g., in either or both of the scale and the parameters.[1] The task of learning is to specify all the unknowns from the known parts of both the densities.

Our *fundamental learning principle* is to make $p(u)$, $q(u)$ be best harmony in a two-fold sense:

- The difference between the resulting $p(u)$, $q(u)$ should be minimized.
- The resulting $p(u)$, $q(u)$ should be of the least complexity.

Mathematically, we use a functional $H(p||q)$ to measure the degree of harmony between $p(u)$ and $q(u)$. When both $p(u)$, $q(u)$ are discrete densities in the form

$$q(u) = \sum_{t=1}^{N} q_t \delta(u-u_t), \quad \sum_{t=1}^{N} q_t = 1 \tag{17}$$

we can simply use the following cross entropy:

$$H(p||q) = \sum_{t=1}^{N} p_t \ln q_t \tag{18}$$

as a typical example of such a measure. The maximization of $H(p||q)$ has two interesting natures:

- *Matching nature* With p fixed, $\max_q H(p||q)$ pushes q toward

$$q_t = p_t, \quad \text{for all } t. \tag{19}$$

- *Least complexity nature* With q fixed, $\max_p H(p||q)$ pushes p toward

$$p(u) = \delta(u-u_\tau), \quad \text{or}$$

$$p_t = \begin{cases} 1, & \text{for } t = \tau \\ 0, & \text{otherwise,} \end{cases} \quad \tau = \arg\max_t q_t. \tag{20}$$

Thus, the maximization of the measure $H(p||q)$ indeed implements the above harmony purpose mathematically. It can also be understood from the fact that $\max_q H(p||q)$ leads to $H(p||p)$ when p is free without other constraint. Clearly, $-H(p||p)$ is the entropy of p, which is a typical complexity measure for a stochastic model p. The further maximization of $H(p||p)$ or equivalently minimization of $-H(p||p)$ pushes $p(u)$ toward (20) when p is also free. The density form of (20) is the simplest from probabilistic view since it simply says that $u = u_\tau$ in probability one, and the entropy $-H(p||p)$ becomes its smallest value 0.

To extend (18) to cover the cases that $q(u)$ is a continuous density, we consider a set $\mathbf{u} = \{u_t\}_{t=1}^N$ of samples that comes independently and identically from $q(u)$ and we form a discrete density $\hat{q}(u)$ as in (17) via the following normalization:

$$\hat{q}_t = q(u_t)/z_q, \quad z_q = \sum_{t=1}^{N} q(u_t). \tag{21}$$

When $q(u)$ is a discrete density in (17), we have

$$z_q = \delta_u(0), \quad \hat{q}_t = q_t, \quad \text{thus,} \quad \hat{q}(u) = q(u). \tag{22}$$

Putting (21) into (18), we have $H(p||q) = \sum_{t=1}^N p_t \ln q(u_t) - \ln z_q$. Similarly, we can also approximate a continuous $p(u)$ and get $\sum_{t=1}^N (p(u_t)/z_q)\ln q(u_t)$. When $\mathbf{u} = \{u_t\}_{t=1}^N$ comes independently and identically from $p(u)$ in a large enough size N, we can further smooth each point u_t by a hyper-cubic bin with a small volume h^k for a small $h > 0$ and regard $p(u_t)h^k$ as the probability that u falls in this bin. Then, $\sum_{t=1}^N p(u_t)h^k = 1$ holds approximately for an appropriate

TABLE I
INDEPENDENT DENSITY AND INDEPENDENT CONDITIONAL DENSITY

$q(y|\xi) = \prod_{j=1}^{k} q(y^{(j)}|\xi_j, \theta_j)$,

(a) It is an independent density when y is irrelevant to ξ, ξ_j.

(b) It is generally a conditional independent density,

with each $q(y^{(j)}|\xi_j, \theta_j)$ being

(i) either a simple one variable density,

e.g., with $\hat{y}^{(j)} = d_j\xi_j + m_j$ we have

Gaussian $\quad q(y^{(j)}|\xi_j, \theta_j) = G(y^{(j)}|\hat{y}^{(j)}, \sigma_j^2)$

Bernoulli $\quad q(y^{(j)}|\xi_j, \theta_j) = s(\hat{y}^{(j)})^{y^{(j)}}[1 - s(\hat{y}^{(j)})]^{1-y^{(j)}}$.

(ii) or a compound one variable density in a finite mixture:

$q(y^{(j)}|\xi_j, \theta_j) = \sum_{r=1}^{\kappa_{j,r}} \alpha_{j,r} q(y^{(j)}|\xi_j, \theta_{j,r})$,

$\sum_{r=1}^{\kappa_{j,r}} \alpha_{j,r} = 1, \ \alpha_{j,r} \geq 0$,

$q(u^{(j)}|\xi_j, \theta_{j,r})$ is a simple density with a known structure,

e.g., with $\hat{y}^{(j,r)} = d_{j,r}\xi_j + m_{j,r}$ we have

Gaussian $\quad q(y^{(j)}|\xi_j, \theta_{j,r}) = G(y^{(j)}|\hat{y}^{(j,r)}, \sigma_{j,r}^2)$

Bernoulli $\quad q(y^{(j)}|\xi_j, \theta_{j,r}) = s(\hat{y}^{(j,r)})^{y^{(j)}}[1 - s(\hat{y}^{(j,r)})]^{1-y^{(j)}}$.

Note: $s(r) = \frac{1}{1+e^{-r}}$ is a scalar function. Moreover, in case (ii), $q(y^{(j)}|\xi_j, \theta_j)$ actually relies on a scale $\kappa_{j,r}$ that is omitted for simplicity but implied in the number of parameters in θ_j.

[1]It can be observed more clearly in Section II-B, where $p(u), q(u)$ denote two different structures of a same joint density.

volume $h^k \approx 1/\sum_{t=1}^{N} p(u_t) = 1/z_q$. We further have $\sum_{u_t} p(u_t)h^k \ln q(u_t) \approx \int p(u) \ln q(u)\, du$ as $h \to 0$ and get the following general form of the harmony measure:

$$H(p||q) = \int p(u) \ln q(u)\, du - \ln z_q, \quad z_q = \sum_{t=1}^{N} q(u_t) \quad (23)$$

which consists of the continuous cross entropy as the first term that accounts for learning with the large size of samples and the second term that accounts for the effect of a finite size of samples.

Furthermore, we can get two typical forms for the implementation of (23).

First, with $p(u)$ given by (16), (23) becomes

$$H(p||q) = \sum_{t=1}^{N} \ln q(u_t) - \ln z_q, \quad z_q = \sum_{t=1}^{N} q(u_t) \quad (24)$$

which applies to both the case that $q(u)$ is a discrete density and that $q(u)$ is a continuous density.

Second, for a continuous density $q(u)$, another choice of $p(u)$ is the Parzen window estimate

$$p(u) = p_h(u), \quad p_h(u) = \frac{1}{N} \sum_{t=1}^{N} G(u|u_t, h^2 I) \quad (25)$$

which returns to $p_0(u)$ as $h \to 0$. That is, $p_h(u)$ is a smoothed modification of $p_0(u)$ by using a Gaussian kernel $G(u|0, h^2 I)$ with mean zero and covariance $h^2 I$ to blur out each impulse at u_t.

Since we desire that the difference between $p(u)$ and $q(u)$ is smallest, it is justified to impose a weak constraint

$$\sum_{t=1}^{N} p(u_t) \approx \sum_{t=1}^{N} q(u_t) = z_q. \quad (26)$$

With this constraint, we can approximately get

$$z_q \approx \frac{z_q^N(h,k)}{N(2\pi h^2)^{k/2}}, z_q^N(h,k) = \sum_{\tau=1}^{N} \sum_{t=1}^{N} e^{-0.5\frac{\|u_t - u_\tau\|^2}{h^2}}. \quad (27)$$

From (25) and (27), (23) becomes

$$H(p||q) = \int p_h(u) \ln q(u)\, du + 0.5k \ln(2\pi h^2) - \ln z_q^N(h,k) + \ln N. \quad (28)$$

The first term encourages the best fitting of $q(u)$ to $p_h(u)$. The smaller is the h, this fitting is more loyal to empirical samples. However, the second term discourages that h becomes too small for avoiding over-fitting on a finite number of samples, especially when the data dimension k is high. The third term balances the second term for avoiding an over-action. Finally, the term $\ln N$ says that a large number N is preferred. In this paper, we only consider the case that N is given and thus the constant $\ln N$ can be neglected. However, (28) can also be used to those extended studies on how many samples are needed to achieve a desired harmony performance.

Based on (24) or (28), we describe a mathematical implementation of harmony learning as follows:

$$\max_{\theta, \mathbf{k}} H(\theta, \mathbf{k}), \quad H(\theta, \mathbf{k}) = H(p||q) \quad (29)$$

where θ consists of all the unknown parameters, and $\mathbf{k}$ consists of unknown parts of scales in $p(u)$, $q(u)$. In the case of (28), the smoothing parameter h is also included in θ, which is thus denoted as $\theta_h = \{\theta, h\}$.

It can be observed that the first term of $H(p||q)$ is actually the likelihood:

$$L(\theta) = \frac{1}{N} \sum_{t=1}^{N} \ln q(u_t|\theta). \quad (30)$$

Thus, if we roughly regard that z_q is approximately irrelevant to θ, $\max_\theta H(p||q)$ becomes equivalent to the conventional ML learning on $q(u|\theta)$. This implementation of harmony learning (29) is made directly based on a sample set $\mathbf{u} = \{u_t\}_{t=1}^{N}$ and thus is also referred as *empirical learning*.

Generally, (24) and (28) provide us two types of the regularized ML learning as follows:

(a) Considering z_q in (23), $\max_\theta H(p||q)$ consists of the ML learning plus a regularization that prevents $q(u|\theta)$ to over-fit the data set $\mathbf{u}$ of a finite size. This point can be better observed by comparing the gradients:

$$\begin{aligned} \nabla_\theta L(\theta) &= Gd(\gamma_t)|_{\gamma_t = 1/N} \\ \nabla_\theta H(p||q) &= Gd(\gamma_t)|_{\gamma_t = (1/N) - \tilde{q}(u_t|\theta)} \\ Gd(\gamma_t) &= \sum_t \gamma_t \nabla_\theta \ln q(u_t|\theta) \\ \tilde{q}(u_t|\theta) &= q(u_t|\theta) \Big/ \sum_\tau q(u_\tau|\theta). \end{aligned} \quad (31)$$

That is, a delearning is added to learning on each sample and the delearning step size is proportional to the degree of fitting on the sample by the current model. The implementation (24) of the harmony learning (29) is made with a normalization term z_q in effect to avoid over-fitting on a data set $\mathbf{u}$ of finite size. Thus, the implementation (24) of the harmony learning (29) is called *normalization learning*.

(b) Considering (28), in help of the Taylor expansion of $\ln q(u|\theta)$ around u_t up to the second order, we have

$$\begin{aligned} &\int G(u|u_t, h^2 I) \ln q(u|\theta)\, du \\ &\quad \approx \ln q(u_t|\theta) + 0.5h^2 \operatorname{Tr}[H_q(u_t|\theta)] \\ &H_q(u_t|\theta) = \frac{\partial^2 \ln q(u|\theta)}{\partial u \partial u^T}. \end{aligned} \quad (32)$$

Then, $\max_\theta H(p||q)$ becomes equivalent to

$$\begin{aligned} &\max_{\theta_h} L_S(\theta_h), \\ &L_S(\theta_h) = 0.5k \ln(2\pi h^2) - \ln z_q^N(h,k) \tilde{L}_S(\theta) \\ &+\tilde{L}_S(\theta_h) = \frac{1}{N} \sum_t \int G(u|u_t, h^2 I) \ln q(u|\theta)\, du \\ &\tilde{L}_S(\theta_h) \approx L(\theta) + 0.5h^2 \lambda_q(\theta) \\ &\lambda_q(\theta) = \frac{1}{N} \sum_t \operatorname{Tr}[H_q(u_t|\theta)] \end{aligned} \quad (33)$$

which regularizes the ML learning via smoothing $\ln g(u_t|\theta)$ in the near-neighbor of u_t and thus is called

data smoothing regularization. So, the implementation (28) of the harmony learning (29) is called *data smoothing learning*. Furthermore, it follows from the approximation in $\tilde{L}_S(\theta)$ that this data smoothing is closely related to Tikhonov-type regularization [58], [29], [9].

We can implement (33) by alternatively repeating the following two steps.

Step 1)

$$\text{fix } h, \text{ get } \theta^{\text{new}} = \theta^{\text{old}} + \eta\delta\theta.$$

Step 2)

$$\begin{aligned}&\text{fix } \theta, \text{ get } h^{\text{new}}\\ &= h^{\text{old}} + \eta\delta h, \delta h = \frac{k}{h} + h\lambda_q(\theta) - \frac{d\ln z_q^N(h,k)}{dh} \quad \text{or}\\ &h^{\text{new2}}\\ &\approx \frac{2h_0^2}{1+\sqrt{1+4h_0^2k^{-1}\lambda_q(\theta)}}\\ h_0^2 &= \frac{1}{k}\sum_{\tau=1}^{N}\sum_{t=1}^{N}\gamma_{t,\tau}\|u_t-u_\tau\|^2, \gamma_{t,\tau} = \frac{e^{\frac{-0.5\|u_t-u_\tau\|^2}{h^{\text{old2}}}}}{z_q(h^{\text{old}},k)}\end{aligned} \tag{34}$$

where $\eta > 0$ is a step size, and $\delta\theta$ is an ascend direction of $\tilde{L}_S(\theta_h)$ or $L(\theta)+0.5h^2\lambda_q(\theta)$. The alternative solution is given by solving the critical equation $\delta h = 0$ with approximately $\frac{d\ln z_q^N(h,k)}{dh} = \frac{kh_0^2}{h^3}$.

The fact that the maximization of the continuous cross entropy $\max_\theta \int p_0(u)\ln q(u|\theta)\,du$ leads to the ML learning (30) are well known in the literature. However, the above two implementations of the matching nature (19) have been rarely studied. More interestingly, the least complexity nature (20) has been regarded as being useless in the conventional sense of learning, but in the sequel we will show that it is this least complexity nature (20) that takes an essential role in learning on the following BYY system.

B. BYY System and Harmony Learning

We consider $u = (x, y)$ with $x \in X$ observable and $y \in Y$ invisible as shown in Fig. 1.

On one hand, we can interpret that each x_t is generated from an invisible inner representation y_t via a backward density[2] $q(x|y)$. The mapping from Y to X by $q(x|y)$ can be understood from two perspectives. One is sample-to-sample mapping $y_t \to x_t$ in three choices as given in Fig. 1. The other is a generative model

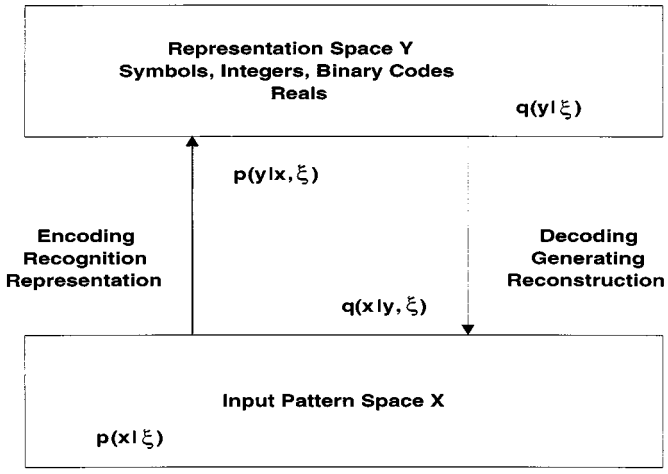
$$q(x) = \int q(x|y)q(y)\,dy \tag{35}$$

that maps from an inner density $q(y)$ in a structure that is designed according to the learning tasks.

On the other hand, we can interpret that each x_t is represented as being mapped into an invisible inner representation y_t via a forward path $p(y|x)$. Again, the mapping from X to Y by $p(y|x)$ can be understood similarly from either the sample-to-sample mapping $x_t \to y_t$ in three choices as shown in Fig. 1 or a representative model

$$p(y) = \int p(y|x)p(x)\,dx \tag{36}$$

that matches the inner density $q(y)$ in a specific structure.



Encoding
Stochastic: randomly pick y by $p(y|x,\xi)$
maximum posteriori: $y = \text{argmax}_y\, p(y|x,\xi)$
regression: $E[y|\xi]$ under $p(y|x,\xi)$

Decoding
Stochastic: randomly pick x by $q(x|y,\xi)$
maximum posteriori: $x = \text{argmax}_x\, q(x|y,\xi)$
regression: $E[x|\xi]$ under $q(x|y,\xi)$

Fig. 1. BYY learning system.

The above two perspectives reflect the two types of Bayesian decomposition of the joint density $q(x|y)q(y) = q(x,y) = p(x,y) = p(x)p(y|x)$ on $X \times Y$. Without any constraints, the two decompositions should be theoretically identical. However, in our above consideration, the four components $p(y|x), p(x)$, $q(x|y)$, $q(y)$ are subject to certain structural constraints. Thus, we usually have two different but complementary Bayesian representations:

$$p(x,y) = p(y|x)p(x), \quad q(x,y) = q(x|y)q(y) \tag{37}$$

which, as discussed in the original paper [73], [80] compliments to the famous Chinese ancient Ying-Yang philosophy with $p(x,y)$ called Yang model that represents the observation space (or called Yang space) by $p(x)$ and the forward pathway (or called Yang pathway) by $p(y|x)$, and with $q(x,y)$ called Ying model that represents the invisible state space (or Ying space) by $q(y)$ and the Ying (or backward) pathway by $q(x|y)$. Thus, such a pair of Ying-Yang models is called BYY system.

With $p(x)$ given by the observed data set $\mathbf{x} = \{x_t\}_{t=1}^N$, the learning task on a BYY system consists of specifying all the aspects of $p(y|x)$, $q(x|y)$, $q(y)$. First, we need to design a combination of structures for $p(y|x)$, $q(y)$, $q(x|y)$ and such a combination is referred as a system architecture. Specifically, $q(y)$

[2]The notation ξ in Fig. 1 will be explained in Section II-E. At this moment, we can simply ignore the existence of ξ.

is always in a parametric structure, with the format of y indicating the inner representation form and the structure of $q(y)$ describing the detailed inner representation. Moreover, the system architecture is featured by the structures of $p(y|x)$, $q(x|y)$, as will be further described in Section II-C.

Second, we need to specify θ that consists of all the unknown parameters and $\mathbf{k}$ that consists of unknown parts of scales in the system architecture. The task is made by the harmony learning (29) which becomes

$$\max_{\theta,\mathbf{k}} H(\theta,\mathbf{k}), \quad H(\theta,\mathbf{k}) = H(p||q) \tag{38}$$

where $H(p||q)$ is obtained by putting $p(u) = p(x,y) = p(y|x)p(x)$, $q(u) = q(x,y) = q(x|y)q(y)$ into (23). That is, we have

$$H(p||q) = \int p(y|x)p(x)\ln[q(x|y)q(y)]\,dx\,dy - \ln z_q \tag{39}$$

which also have two specific forms corresponding to (24) and (28), respectively.

Corresponding to (24), based on a set $\mathbf{x} = \{x_t\}_{t=1}^N$ of i.i.d. samples, we can get the empirical density $p(x) = p_0(x)$ by (16) and put it into (39), resulting in

$$\begin{aligned} H(\theta,\mathbf{k}) &= H(p||q) = \frac{1}{N}\sum_{t=1}^{N} H_t(p||q) - \ln z_q \\ H_t(p||q) &= \int p(y|x_t)\ln[q(x_t|y)q(y)]\,dy. \end{aligned} \tag{40}$$

Moreover, as will be shown in Section II-C, due to the least complexity nature (20) in the harmony learning (38), $p(y|x)$ of two typical structures will be pushed into

$$\begin{aligned} p(y|x_t) &= \delta(y - F(x_t)) \text{ at each } x_t \text{ and } F(x_t) \\ &\quad \text{is a function of } x_t. \end{aligned} \tag{41}$$

Thus, the above $H(\theta,\mathbf{k})$ further becomes

$$\begin{aligned} H(\theta,\mathbf{k}) &= \frac{1}{N}\sum_{t=1}^{N} H_t(p||q) - \ln z_q, \\ H_t(p||q) &= \ln q(x_t|y_t) + \ln q(y_t), \\ y_t &= F(x_t), \quad z_q = \sum_{t=1}^{N} q(x_t|y_t)q(y_t). \end{aligned} \tag{42}$$

Similar to (31), this implementation of the harmony learning (38) is called *normalization learning*. If we only consider the first part by regarding z_q to be irrelevant to learning, it becomes *empirical learning* that directly bases on the samples $\mathbf{x} = \{x_t\}_{t=1}^N$. This empirical learning on the BYY system is the counterpart of the ML learning (30).

Corresponding to (28), for the cases that both $q(x|y)$ and $q(y)$ are continuous densities, we use the Parzen estimate $p(x) = p_{h_x}(x)$ by (25) and consider the following soften version of (41):

$$\begin{aligned} p(y|x_t) &= G(y|F(x_t), h_y^2 I) \text{ at each } x_t \\ &\quad \text{and } F(x_t) \text{ is a function of } x_t. \end{aligned} \tag{43}$$

Putting them into (39), we get

$$\begin{aligned} H(\theta_h,\mathbf{k}) &= H_h(p||q) = \frac{1}{N}\sum_{t=1}^{N} H_{h,t}(p||q) - \ln z_q \\ h &= \{h_x, h_y\} \\ H_{h,t}(p||q) &= \int G(x|x_t, h_x^2 I)G(y|F(x_t), h_y^2 I) \\ &\quad \cdot \ln[q(x|y)q(y)]\,dx\,dy. \end{aligned} \tag{44}$$

Moreover, we consider the constraint (26), with $y_t = F(x_t)$ and

$$z_q^N(h,k) = \sum_{\tau=1}^{N}\sum_{t=1}^{N} e^{-0.5[\frac{||(x_t - x_\tau)||^2}{h_x^2} - \frac{||(y_t - y_\tau)||^2}{h_y^2}]} \tag{45}$$

we have

$$z_q \approx \frac{z_q^N(h,k)}{N(2\pi h^2)^{d_x/2}(2\pi h^2)^{k/2}}, d_x \text{ is the dimension of } x. \tag{46}$$

Furthermore, similar to (32), we let $\ln q(x|y)$, $\ln q(y)$ to be approximated by its Taylor expansion around x_t, $y_t = F(x_t)$ up to the second order and notice that

$$\begin{aligned} &\int G(x|x_t, h_x^2 I)G(y|F(x_t), h_y^2 I)(x - x_t)(y - F(x_t))^T\,dx\,dy \\ &\quad = \int G(x|x_t, h_x^2 I)(x - x_t) \\ &\qquad \cdot \left[\int G(y|F(x_t), h_y^2 I)(y - F(x_t))^T\,dy\right] dx = 0, \end{aligned}$$

it follows from (44) and (46) that

$$\begin{aligned} H(\theta_h,\mathbf{k}) &= H_h(p||q) \\ &= \frac{1}{N}\sum_{t=1}^{N} H_{h,t}(p||q) + 0.5d_x\ln(2\pi h_x^2) \\ &\quad + 0.5k\ln(2\pi h_y^2) - \ln z_q^N(h,k) + \ln N, \\ H_{h,t}(p||q) &= \ln q(x_t|y_t) + \ln q(y_t) + 0.5h_x^2 \\ &\quad \cdot \operatorname{Tr}[H_q^x(x_t|y_t)] + 0.5h_y^2 \operatorname{Tr}[H_q^y(x_t|y_t) + H_q(y_t)], \\ y_t &= F(x_t), \quad H_q^x(x|y) = \frac{\partial^2 \ln q(x|y)}{\partial x \partial x^T}, \\ H_q^y(x|y) &= \frac{\partial^2 \ln q(x|y)}{\partial y \partial y^T}, \quad H_q(y) = \frac{\partial^2 \ln q(y)}{\partial y \partial y^T}. \end{aligned} \tag{47}$$

A discussion similar to that after (28) can be made on each item of $H(\theta_h,\mathbf{k})$, as well as on the role of the size N. Moreover, similar to (33), this implementation of the harmony learning (38) is called *data smoothing learning*. It degenerates back to *empirical learning* if we only consider the part $(1/N)\sum_{t=1}^{N} H_{h,t}(p||q)$ with $h_x = 0$, $h_y = 0$.

C. Three Typical Architectures

A BYY system architecture, consisting of $q(y)$, $p(y|x)$, $q(x|y)$, is featured by the specific structures of $p(y|x)$ and $q(x|y)$. Each of $p(y|x)$ and $q(x|y)$ can be either parametric or structure free. We say $p(u|v)$ is structural free in the sense that $p(u|v)$ can be any function that satisfies $\int p(u|v) = 1$,

$p(u|v) \geq 0$. In learning, a structure-free density is actually specified in terms of other parametric structures. An architecture with both $p(y|x)$, $q(x|y)$ being structure-free is meaningless since they are no longer able to be specified via learning. Therefore, there remain the following three choices for a meaningful BYY architecture:

- Backward architecture (B-architecture): $p(y|x)$ is structure-free and $q(x|y)$ is parametric.
- Forward architecture (F-architecture): $q(x|y)$ is structure-free and $p(y|x)$ is parametric.
- Bi-directional architecture (BI-architecture): both $p(y|x)$, $q(x|y)$ are parametric.

In a B-architecture, due to the least complexity nature (20), the harmony learning (38) pushes a free $p(y|x)$ into the least complexity form (41) with

$$y_t = F(x_t) = \arg\max_y [q(x_t|y)q(y)]. \tag{48}$$

Thus, we get (42). Moreover, due to the matching nature (19), the least complexity form (41) will further push $q(x|y)$ and $q(y)$ to the least complexity. This nature can also be understood by observing that maximizing $H_t(p||q)$ in (42) consists of both maximizing $\ln q(x_t|y_t)$ and maximizing $\ln q(y_t)$ that push $q(x|y)$, $q(y)$ to be as close as possible to their least complexity forms $q(x_t|y_t) = \delta(x_t - g(y_t))$ and $q(y_t) = \delta(y_t - y_0)$.

The process in (48) can also be understood as implementing a competition coordinately based on both $q(x_t|y)$ and $q(y)$, with $q(x_t|y)$ representing the regression between the observation x_t and its inner representation y, and with $q(y)$ representing the preference of this inner representation. Thus, such a type of competition is called *coordinate competition* [66]. It is also called *a posteriori* competition because the resulted $p(y|x)$ is actually winner-take-all (WTA) based on the posteriori probability $q(x_t|y)q(y)/\int q(x_t|y)q(y)\,dy$ and thus y_t is also called a maximum *a posteriori* probability (MAP) estimate.

However, it is well known that a greedy WTA competition will create local optimal solutions or even a bad solution, which is usually referred as the "dead unit" problem in the classical competitive learning [23]. Such a local optimal problem is solved from several aspects in the harmony learning. As shown in (31), one aspect is that the term $-\ln z_q$ introduces a delearning effect to the postcompetition learning on $q(x_t|y)$ and $q(y)$. Other aspects can be observed in a BI-architecture.

In a BI-architecture, $p(y|x)$ is not free but in a parametric structure. In this case, due to the least complexity nature (20), the harmony learning (38) in implementation via (40) will push $p(y|x)$ again toward to (48) subject to certain structural constraint. This motivates us to design

$$p(y|x) = \sum_{j=1}^{k_f} \gamma(j|x)\delta(y - f_j(x,\theta_{f_j})), \qquad 0 \leq \gamma(j|x) \leq 1$$
$$\sum_{j=1}^{k_f} \gamma(j|x) = 1 \tag{49}$$

which is a piecewise-structure that stochastically selects among one of k_f deterministic functions with probabilities $\gamma(j|x_t)$, $j = 1, \cdots, k_f$. In this case, maximizing $H(\theta, k)$ with respect to $\gamma(j|x_t)$ leads to (42) again but with

$$y_t = F(x_t) = f_{j_*}(x_t, \theta_{f,j_*})$$
$$\gamma(j|x_t) = \begin{cases} 1, & j = j_* \\ 0, & \text{otherwise} \end{cases} \tag{50}$$

where j_* is the winner of the following constrained competition:

$$j_* = \arg\max_j [q(x_t|y_{tj})q(y_{tj})], \quad y_{tj} = f_j(x_t, \theta_{f,j_*}) \tag{51}$$

which not only makes the greedy WTA competition (48) be more "conscience" but also saves computing cost considerably. However, when the structure of $p(y|x)$ is too simple to describe the relation $F(x_t)$ in (48), the resulted BYY system may result in a poor performance on modeling $\mathbf{x} = \{x_t\}_{t=1}^N$.

The implementation of harmony learning via data smoothing (47) is also an example that regularizes the WTA competition (48) via the parametric structure (43) that adds "conscience" to both the WTA competition (48) by introducing noise and the postcompetition learning on $q(x_t|y)$ and $q(y)$ via $H(\theta_h, \mathbf{k})$ given in (47). For simplicity, we can also let $F(x_t)$ given by (50). Actually, (43) with $F(x_t)$ given by (50) provides a $p(y|x)$ that has a piecewise-structure which stochastically selects according to $\gamma(j|x_t)$ in (50), among functions of $f_j(x_t, \theta_{f,j})$ under a Gaussian noise with variance $h_y^2 I$.

Another strategy is to regard the learning problem as a typical optimization for finding the global optimal solution in the case there are many local optimal solutions. We can use one of existing classical global optimization techniques for this purpose, e.g., during the implementation of harmony learning via data smoothing (47) we can use the popular simulated annealing technique [38] via letting h_x, h_y to start at some given values and then gradually to reduce to zero during the learning (47).

In a F-architecture, $q(x|y)$ is free and we simply set $q(x|y) = p_0(x)$ by (16). It follows from (42) that $H_t(p||q) = \ln \delta_x(0) + \ln q(y_t)$ still remains meaningful because the infinite term $\ln \delta_x(0)$ is cancelled out by the one from $z_q = \ln[\delta_x(0) \sum_{t=1}^N q(y_t)]$. However, in this case, it is nonsense to consider whether the inner representation y_t can reconstruct x_t well. Instead, we only have the constraint that y_t is independent in components. Therefore, the scale indeterminacy similar to (4) cannot avoided, which has two consequences. One is that we need to prefix the value of k because it becomes nonsense to consider model selection in a F-architecture. The other consequence is that we need to prefix the covariance matrix associated with each regression $y_{tj} = f_j(x, \theta_{f_j})$ to a given diagonal matrix D_j, otherwise the learning process may not converge because of the scale indeterminacy of D_j. As a result, from (40) we have

$$H(\theta, \mathbf{k}) = \frac{1}{N}\sum_{t=1}^N \ln q(y_t) - \ln z_y,$$
$$z_y = \sum_{t=1}^N q(y_t), \text{ subject to } \Sigma_{yj} = D_j, \text{ for } j = 1, \cdots, k_f$$
$$\Sigma_{yj} = E[(y_j - Ey_j)(y_j - Ey_j)^T], \quad y_j = f_j(x, \theta_{f_j})$$
$$y_t \text{ is given by (50) with } j_* = \arg\max_j q(y_{tj}). \tag{52}$$

D. Parameter Learning and Model Selection: Parallel versus Sequential

From the above discussions, we observe that the parameter set θ_h or θ always contains the unknown parameters of $q(y)$ plus the unknown parameters in one or both of $p(y|x)$ and $q(x|y)$, with no parameter from either $p(y|x)$ for a B-architecture or $q(x|y)$ for a F-architecture. The scale set $\mathbf{k}$ is featured by $q(y)$. For a $q(y)$ given by Table I, the dimension k of y indicates the scale of the representation space and $\{\kappa_{j,r}\}$ further describes the complexity of the detailed inner representations. If k, $\{\kappa_{j,r}\}$ are too small, we could not get good inner representations due to the limited representing capacity. On the other hand, if they are too large, it is difficult to form compact representations and thus the modeling performance of the BYY system is also poor. With appropriate k, $\{\kappa_{j,r}\}$, the roles of $p(y|x)$ and $q(x|y)$ are setting up a bi-directional mapping that best preserves information. For this purpose, the structures of $q(x|y)$ and $p(y|x)$ are designed with a mapping capacity large enough, nevertheless the extra capacity is further minimized by the least complexity nature (20).

Moreover, parameter learning for θ_h or θ may imply making model selection for $\mathbf{k}$. As discussed before, due to the least complexity nature (20), the implementation of (38) via normalization learning (42) will push $q(y|\theta)$ toward the least complexity form $\delta(y-y_0)$. Specifically, if the component $y^{(j)}$ is redundant in representing the observed data, $q(y^{(j)}|\theta_j)$ will be forced toward to $\delta(y^{(j)}-y_0)$ that can be reached by setting all of $\alpha_{j,r}$ except one to zero and forcing the variance of $q(y^{(j)}|\theta_j)$ to zero, when the structure of $q(y|\theta)$ is given in Table I with fixed k, $\{\kappa_{j,r}\}$. For example, we get $G(y^{(j)}|m^{(j)},\sigma^{(j)2}) = \delta(y^{(j)}-m^{(j)})$ when $\sigma^{(j)2}=0$. In fact, $q(y^{(j)}|\theta_j) = \delta(y^{(j)}-y_0)$ means that the jth dimension is removed effectively since $y^{(j)}$ becomes a constant. Thus, the dimension k is effectively reduced by one. It can be observed from (42) that $H(\theta,\mathbf{k})$ will remain bounded even when $q(y^{(j)}|\theta_j) = \delta(y^{(j)}-y_0)$. As an infinite $\ln\delta_{y^{(j)}}(0)$ from $H_t(p||q)$, we will simultaneously get $-\ln\delta_{y^{(j)}}(0)$ from $\ln z_q$, and thus the effects of two infinite terms are cancelled.

While for the harmony learning (38) implemented via smoothing (47), the least complexity nature (20) will not lead to (41), since the parameter h_y^2 is bounded from 0, and thus $q(y|\theta)$ will not become the least complexity form $\delta(y-y_0)$. As a result, the above automated model selection will not occur. Even so, the least complexity nature (20) will still push h_y as well as the variance of each $q(y^{(j)}|\theta_j)$ to be as small as possible. That is, $q(x|y)$ and $q(y)$ will still be pushed toward their minimum complexity subject to the structure (43).

With the above understanding, we can get two typical procedures for implementing the harmony learning (38) on a B-architecture or a BI-architecture with $H(\theta,\mathbf{k})$ given by either (42) or (47).

1) Parameter Learning with Automated Model Selection: We set the scales in $\mathbf{k}$ large enough and implement the harmony learning (38) by

$$\max_{\theta} H(\theta), \quad H(\theta) = H(\theta,\mathbf{k}). \tag{53}$$

The least complexity nature (20) as well as its specific forms (48) and (51) will push each $q(y^{(j)}|\theta_j)$ toward the form $\delta(y^{(j)}-m^{(j)})$ and even reach it when there is no constraint to block it. This nature can also be more directly observed from (42) and (52), where maximizing $\ln q(y_t) = \sum_{j=1}^{k} \ln q(y^{(j)}|\theta_j)$ pushes each $q(y^{(j)}|\theta_j)$ toward the form $\delta(y^{(j)}-m^{(j)})$. If one of them is reached, it is effectively equivalent to reducing k to $k-1$. In other words, model selection is made automatically in parallel to parameter learning. Also, the least complexity nature (50) will force $p(y|x)$ to be used in a way of being effectively the least complexity. In turn, the matching nature (19) will also force $q(x|y)$ to be effectively of the least complexity.

2) Parameter Learning Followed by Model Selection: In the cases that k, $\{\kappa_{j,r}\}$ are not large enough or in the data smoothing learning (47), where the parameter h_y^2 is bounded from 0 and thus $q(y^{(j)}|\theta_j)$ is not able to reach $\delta(y^{(j)}-m^{(j)})$. The least complexity nature (20) will still push $q(x|y)$ and $q(y)$ toward the structures of the minimum complexity, but not necessarily lead to the consequence that the scales k, $\{\kappa_{j,r}\}$ are automatically reduced. Thus, as an alternative to making the learning (53) at the scales of k, $\{\kappa_{j,r}\}$ large enough, we can also make parameter learning and model selection sequentially in two steps. With $\{\kappa_{j,r}\}$ prefixed, we enumerate k from a small value incrementally, and at each specific k we perform parameter learning by (53) to get the best parameter value θ^*. Then, we select a best k^* by

$$\min_{k} J(k), \quad J(k) = -H(\theta^*,\mathbf{k}). \tag{54}$$

If there are more than one values of k such that $J(k)$ gets the same minimum, we take the smallest.

Remarks:

1) As discussed at the end of Section II-C, model selection is not applicable to a F-architecture, where we only implement parameter learning (53) with (52) under a prefixed D_j and a given k.
2) If $\max_{\theta,\mathbf{k}} H(\theta,\mathbf{k})$ has only an unique solution, the above two implementations would be equivalent and also both are equivalent to (38). However, in practice, $H(\theta,\mathbf{k})$ may have many local maximums and thus the two implementations may lead to different local optimal solutions. Similar to the features of the conventional parallel computing and sequential computing, the parallel implementation considers a model in a large scale with model selection made in parallel to parameter learning, and thus it is implemented fast in time. While the sequential implementation considers many models incrementally from a small scale to a large one with a lot of computing cost consumed, but there is a minimum waste on using model structure.
3) The idea of finding a minimum complexity structure to implement learning for a better generalization has been well adopted in literature from many perspectives. Typical examples include minimum complexity density estimation [6], MDL theory [48], Bayesian theory [39], VC dimension based generalization theory [60], Tikhonov-type regularization [58], [29], [9], cross validation [55] and AIC [2] as well as its extensions AICB, CAIC, SIC [12]. The harmony learning principle handles the issue from a new perspective that bases on the

least complexity nature (20), featured by its easy in implementation. Its relation to Tikhonov-type regularization has been discussed previously in Section II-A. Its relation to other studies, especially to support vector machine and the VC dimension based generalization theory is referred to [62].

E. Temporal BYY System and Harmony Learning

We further consider observations $\mathbf{x} = \{x_t\}_{t=1}^N$ with temporal relations among samples. Correspondingly, the inner representation is also a temporal series $\mathbf{y} = \{y_t\}_{t=1}^N$. To model the temporal relations, we consider to put $p(u) = p(\mathbf{x}, \mathbf{y})$ and $q(u) = q(\mathbf{x}, \mathbf{y})$ into (23) for implementing learning. But this situation is too complicated to handle and further simplifications are needed.

In [63], [76], and [77], we impose the causal or rationale assumption that x_t, y_t only depend on their values at past $\tau < t$ but not on any future $\tau > t$. With $p(u) = p(\mathbf{x}, \mathbf{y})$ and $q(u) = q(\mathbf{x}, \mathbf{y})$ into (23), we get

$$\begin{aligned} H(p||q) &= \sum_{t=1}^{N} [H_t(p||q) - \ln z_{q,t}] \\ H_t(p||q) &= \int p(\xi_t) H_t(p||q, \xi_t)\, d\xi_t \\ H_t(p||q, \xi_t) &= \int p(y_t|x_t, \xi_t) p(x_t|\xi_t) \\ &\quad \cdot \ln[q(x_t|y_t, \xi_t) q(y_t|\xi_t)]\, dx_t\, dy_t \end{aligned} \tag{55}$$

where ξ_t is a set that consists of all or a part of the samples x_τ, y_τ in the past $\tau < t$.

The above $H_t(p||q)$ involves an integral over ξ_t. To get rid of it, we consider the Taylor expansion of $H_t(p||q, \xi_t)$ around $\hat{\xi}_t = E(\xi_t)$ up to the second order and approximately get

$$\begin{aligned} H_t(p||q) &= H_t(p||q, \hat{\xi}_t) + c_T \operatorname{Tr}[\Sigma_{\xi_t} H_{\xi_t}] \\ H_{\xi_t} &= \left. \frac{\partial H_t(p||q, \xi_t)}{\partial \xi_t \xi_t^T} \right|_{\xi_t = \hat{\xi}_t}. \end{aligned} \tag{56}$$

For simplicity, this paper will focus on the case $c_T = 0$, but all the discussions can be extended by taking the term $\operatorname{Tr}[\Sigma_{\xi_t} H_{\xi_t}]$ in consideration for further adjustments.

By setting $c_T = 0$ and also letting $p(x_t|\xi_t) = \delta(x_t - \overline{x}_t)$ since "$x_t = \overline{x}_t$" happens already and thus is irrelevant to any past sample, it follows from (55) that:

$$\begin{aligned} H(p||q) &= \sum_{t=1}^{N} [H_t(p||q, \hat{\xi}_t) - \ln z_{q,t}] \\ &= N \left[\frac{1}{N} \sum_{t=1}^{N} H_t(p||q, \hat{\xi}_t) - \ln z_q \right] \\ z_q &= \left[\prod_{t=1}^{N} z_{q,t} \right]^{1/N} \\ H_t(p||q, \hat{\xi}_t) &= \int p(y_t|\overline{x}_t, \hat{\xi}_t) \\ &\quad \cdot \ln[q(\overline{x}_t|y_t, \hat{\xi}_t) q(y_t|\hat{\xi}_t)]\, dy_t. \end{aligned} \tag{57}$$

In the special case that samples in $\mathbf{x} = \{\overline{x}_t\}_{t=1}^N$ are i.i.d. with no temporal relation, all the appearances of $\hat{\xi}_t$ can be removed. In this case, we have $z_{q,t} = z_q$ and (57) becomes equivalent to (40). In other words, (57) extends (40) to temporal modeling via taking the past samples $\hat{\xi}_t$ into consideration.

Particularly, further imposing the Markovian assumption that x_t, y_t depend only a finite number of their past sample values, ξ_t can be represented as a vector in a fixed dimension and thus $p(y_t|x_t, \xi_t)$, $q(x_t|y_t, \xi_t)$ and $q(y_t|\xi_t)$ can take regular structures with a fixed number of parameters in each of the structures. In this case, we get a temporal Bayesian Ying-Yang (TBYY) system as a general state space model. Furthermore, under the independence condition on the components of y_t as in (14), not only a unified point of view is obtained on hidden Markov model (HMM), temporal factor analysis (TFA), and temporal independent component analysis (TICA) as well as their application in blind separation of temporal signals, but also adaptive algorithms are developed for implementing TFA, TICA, and a higher order independent HMM, with the corresponding criteria provided for selecting an appropriate number k as the dimension of y_t. The details are referred to [63].

A key difference in (57) from (40) is that at each time t we must already have $\hat{\xi}_t = E(\xi_t)$. Due to this nature, it is difficult to directly implement learning with $H(\theta, \mathbf{k}) = H(p||q)$ on the whole batch of all the N samples. Instead, it is more convenient to make learning recursively at time t, that is, we update

$$\begin{aligned} \theta^{new} &= \theta^{old} + \eta \delta\theta, \text{ with } \delta\theta \text{ being an ascend} \\ &\quad \text{direction of } H_t(\theta, \mathbf{k}) \\ H_t(\theta, \mathbf{k}) &= \frac{1}{t} \sum_{\tau=1}^{t} H_\tau(p||q, \hat{\xi}_\tau) - \ln z_q(t) \\ z_q(t) &= \left[\prod_{\tau=1}^{t} z_{q,\tau} \right]^{1/t} \end{aligned} \tag{58}$$

where $\hat{\xi}_t$ is estimated as t goes. In contrast, there is not such a temporal constraint on learning with (40), where samples in $\mathbf{x}$ are i.i.d. and thus learning at t is independent from ξ_t. We can implement learning (53) with (40) either recursively as in (58) or in a whole batch of all the N samples.

Specifically, $\hat{\xi}_t = E(\xi_t)$ is a set that consists of all of the past samples $E(x_\tau), E(y_\tau)$, $\tau < t$. We simply have $E(x_\tau) = \overline{x}_\tau$. What we need to get is $\hat{y}_\tau = E(y_\tau) = \int y_\tau p(y_\tau)\, dy_\tau$ for all $\tau < t$. It follows that $p(y_\tau) = \int p(\xi_\tau) p(y_\tau|x_\tau, \xi_\tau)\, d\xi_\tau$. To remove the integral over ξ_τ, we approximately let $p(y_\tau|x_\tau, \xi_\tau)$ replaced by its linear expansion around the mean $\hat{\xi}_\tau$, resulting in $p(y_\tau) \approx p(y_\tau|x_\tau, \hat{\xi}_\tau)$. Thus, we get $\hat{\xi}_t$ that consists of all the past samples $\overline{x}_\tau$ and $\hat{y}_\tau$ for all $\tau < t$, with $\hat{y}_\tau$ obtained recursively by

$$\hat{y}_\tau = \int y_\tau p(y_\tau|\overline{x}_\tau, \hat{\xi}_\tau)\, dy_\tau. \tag{59}$$

The current estimate $\hat{y}_t$ can be directly used as $\hat{y}_\tau$ next time, thus no additional effort is needed to get $\hat{y}_\tau$.

In (58), we also need the specific form of $z_{q,t}$. At time t, we have one sample pair $\overline{x}_t$, $\hat{y}_t$ only and thus cannot get $z_{q,t}$ simply as in (42) and (52). To find an alternative way, we observe a special example $q(x_t|y_t, \xi_t) = G(x_t|Ay_t + B\xi_t + \mu, \sigma^2 I)$ for a linear relation $x_t = Ay_t + B\xi_t + \mu + e_t$, where e_t is a Gaussian white noise $G(e_t|0, \sigma^2 I)$, with i.i.d samples of e_t at different times. Though x_t, y_t are time-varying as t goes, both the regression parameters A, B, μ and the density parameter σ^2 in $G(e_t|0, \sigma^2 I)$ are not time-varying, and $q(x_t|y_t, \xi_t) =$

$G(x_t|Ay_t + B\xi_t + \mu, \sigma^2 I)$ is also not time-varying or called stationary. Thus, considering $q(x_t|y_t, \xi_t)$ and $q(y_t|\xi_t)$ that are stationary in this sense, we can use all the past samples $\overline{x}_\tau$ and $\hat{y}_\tau$ for all $\tau \leq t$ to get $z_{q,t}$ in the same way as in (42) and (52). Moreover, we can simply take $z_q(t) = z_{q,t}$ by regarding $z_{q,t} = z_{q,t-1} = \cdots z_{0,t-1}$ since we can use all the samples $\overline{x}_\tau$ and $\hat{y}_\tau$ for all $\tau \leq t$ to reestimate all the past $z_{q,\tau}$ based on the current $q(x_t|y_t, \xi_t)$ and $q(y_t|\xi_t)$.

In the rest of this paper, we focus on (57) with the three typical architectures in Section II-C. Specifically, we use $q(y_t|\hat{\xi}_t)$ as given in Table I with y replaced by y_t and ξ by $\hat{\xi}_t$, which covers the cases of (12) and (14). Also, we use the following conditional finite mixture:

$$q(x_t|y_t, \xi_t) = \sum_{j=1}^{k_g} \beta(j|y_t, \xi_t) q(x_t|y_t, \xi_t, \theta_{g,j}) \qquad 1 \leq \beta(j|y_t, \xi_t) \leq 0, \quad \sum_{j=1}^{k_g} \beta(j|y_t, \xi_t) = 1 \tag{60}$$

which is called the *mixture-of-expert model* [32], [33] since it is a weighted sum of each expert $q(x_t|y_t, \xi_t, \theta_{g,j})$, gated by $\beta(j|y_t, \xi_t)$ that is described by either a softmax model or the alternative gate [66], [74], [81]. Specifically, each $q(x_t|y_t, \xi_t, \theta_{g,j})$ can be any conditional density. Two typical examples are

$$q(x_t|y_t, \xi_t, \theta_{g,j}) = \begin{cases} G(x_t|\hat{x}_t, \Sigma_j), & \text{(a) Gaussian} \\ \prod_{j=1}^{n} [s(\hat{x}_t^{(j)})\delta(x_t^{(j)} - 1) \\ \quad + (1 - s(\hat{x}_t^{(j)}))\delta(x_t^{(j)})] & \text{(b) Bernoulli} \end{cases}$$
$$\hat{x}_t = A_j y_t + B_j \xi_t + \mu_j, \quad s(r) = 1/(1 + e^{-r}). \tag{61}$$

The choice (a) is an additive linear model $x_t = A_j y_t + B_j \xi_t + \mu_j + \varepsilon_j$ with ε_j being independent from y_t, ξ_t and coming from Gaussian with zero mean and covariance Σ_j. The choice (b) describes the case that both x_t, y_t are binary with independent bits.

Moreover, we extend the structure of (49) into

$$p(y_t|x_t, \xi_t) = \sum_{j=1}^{k_f} \gamma(j|x_t, \xi_t) \delta(y_t - f_j(x_t, \theta_{f_j}|\xi_t)) \qquad 0 \leq \gamma(j|x_t, \xi_t) \leq 1, \quad \sum_{j=1}^{k_f} \gamma(j|x_t, \xi_t) = 1. \tag{62}$$

One particular example of $f_j(x_t, \theta_{f_j}|\xi_t)$ is the following post-linear function:

$$f_j(x_t, \theta_{f_j}|\xi_t) = f^\circ(W_j x_t + K_j \xi_t + \mu_j) \tag{63}$$

where $f^\circ(y)$ is a function given as follows:

$$f^\circ(y) = [f(y^{(1)}, \cdots, f(y^{(k)})]^T, \quad y = [y^{(1)}, \cdots, y^{(k)}]^T,$$

$f(r)$ is scalar function, e.g.

$$f(r) = \begin{cases} r, & \text{linear} \\ s(r) = 1/(1 + e^{-r}), & \text{sigmoid.} \end{cases} \tag{64}$$

When $f(r) = r$, we have $f^\circ(y) = y$ and thus $p(y_t|x_t, \xi_t)$ in (62) describes a stochastic piecewise linear functions.

In either a B-architecture or a BI-architecture, similar to (42), we have

$$H_t(\theta, \mathbf{k}) = \frac{1}{t} \sum_{\tau=1}^{t} [\ln q(\overline{x}_\tau|\hat{y}_\tau, \hat{\xi}_\tau) + \ln q(\hat{y}_\tau|\hat{\xi}_\tau)] - \ln z_q(t),$$
$$z_q(t) = \sum_{\tau=1}^{t} q(\overline{x}_\tau|\hat{y}_\tau, \hat{\xi}_\tau) q(\hat{y}_\tau|\hat{\xi}_\tau). \tag{65}$$

In a F-architecture, similar to (52), we get

$$H_t(\theta, \mathbf{k}) = \frac{1}{t} \sum_{\tau=1}^{t} \ln q(\hat{y}_\tau|\hat{\xi}_\tau) - \ln z_q^y(t),$$
$$z_q^y(t) = \sum_{\tau=1}^{t} q(\hat{y}_\tau|\hat{\xi}_\tau)$$
$$\text{subject to } \Sigma_{yj} = E[(\hat{y}_{tj} - Ey_{tj})(\hat{y}_{tj} - Ey_{tj})^T] = D_j, \text{ for } j = 1, \cdots, k_f \tag{66}$$

where each D_j is an arbitrarily prefixed diagonal matrix, with positive diagonal elements.

In both (65) and (66), $\hat{y}_t$ is obtained in a way similar to (48) and (50). That is, we have

$$\hat{y}_t = F(\overline{x}_t, \hat{\xi}_t) = \begin{cases} \arg\max_y [q(\overline{x}_t|y, \hat{\xi}_t) q(y|\hat{\xi}_t)] \\ \quad \text{for a B-architecture} \\ f^\circ(W_{j_*} \overline{x}_t + K_{j_*} \hat{\xi}_t + \mu_{j_*}) \\ \quad \text{for a BI- \& F-architecture} \end{cases}$$
$$j_* = \begin{cases} \arg\max_j [q(\overline{x}_t|\hat{y}_{tj}, \hat{\xi}_t) q(\hat{y}_{tj}|\hat{\xi}_t)] \\ \quad \text{for a BI-architecture} \\ \arg\max_j [q(\hat{y}_{tj}|\hat{\xi}_t)] \\ \quad \text{for a F-architecture} \end{cases}$$
$$\hat{y}_{tj} = f^\circ(W_j \overline{x}_t + K_j \hat{\xi}_t + \mu_j). \tag{67}$$

With $H_t(\theta, \mathbf{k})$ given by (65) or (66), we can recursively implement parameter learning (53) via (58). Particularly, on a B-architecture or a BI-architecture, model selection is made either automatically during parameter learning or via selecting a best k^* by (54) with k enumerated incrementally as (58) is implemented over all of N samples at each k.

Furthermore, for a B-architecture or a BI-architecture, we can also implement the harmony learning via data smoothing learning (56) with $c_t = 0$ in help of the temporal extension of (47).

With $F(\overline{x}_t, \hat{\xi}_t)$ given by (67), by considering

$$p(x_t|\xi_t) = G(x_t|\overline{x}_t, h_x^2 I),$$
$$p(y_t|\overline{x}_t, \xi_t) = G(y_t|F(\overline{x}_t, \hat{\xi}_t), h_y^2 I) \tag{68}$$

it follows from (55) that instead of (57) we get

$$\frac{1}{t} H_h(p||q) = \frac{1}{t} \sum_{\tau=1}^{t} H_{h,t}(p||q, \hat{\xi}_t) - \ln z_q(t)$$
$$h = \{h_x, h_y\}$$
$$H_{h,t}(p||q, \hat{\xi}_t) = \int G(x|x_t, h_x^2 I) G(y_t|F(\overline{x}_t, \hat{\xi}_t), h_y^2 I) \cdot \ln[q(x_t|y_t, \hat{\xi}_t) q(y_t|\hat{\xi}_t)] \, dx_t dy_t. \tag{69}$$

For a similar reason discussed above, we can simply take $z_q^{(t)} = z_{q,t}$ that is then approximately given by (46) under the constraint (26).

Following a similar process from (44) to (47), from (69) we get

$$\begin{aligned}
H_t(\theta_h,\mathbf{k}) &= \frac{1}{t}\sum_{\tau=1}^{t} H_{h,t}(p||q,\hat{\xi}_t) + 0.5d_x\ln(2\pi h_x^2) \\
&\quad + 0.5k\ln(2\pi h_y^2) - \ln z_q^t(h,k) + \ln t, \\
\hat{y}_t &= F(\overline{x}_t,\hat{\xi}_t) \\
H_{h,t}(p||q,\hat{\xi}_t) &= \ln q(\overline{x}_\tau|\hat{y}_\tau,\hat{\xi}_\tau) + \ln q(\hat{y}_\tau|\hat{\xi}_\tau) \\
&\quad + 0.5h_x^2\ \mathrm{Tr}[H_q^x(\overline{x}_t|\hat{y}_t,\hat{\xi}_t)] \\
&\quad + 0.5h_y^2\ \mathrm{Tr}[H_q^y(\overline{x}_t|\hat{y}_t,\hat{\xi}_t) + H_q(\hat{y}_t|\hat{\xi}_t)] \\
H_q^x(x_t|y_t,\xi_t) &= \frac{\partial^2\ln q(x_t|y_t,\xi_t)}{\partial x_t\partial x_t^T} \\
H_q^y(x_t|y_t,\xi_t) &= \frac{\partial^2\ln q(x_t|y_t,\xi_t)}{\partial y_t\partial y_t^T} \\
H_q(y_t|\xi_t) &= \frac{\partial^2\ln q(y_t|\xi_t)}{\partial y_t\partial y_t^T}.
\end{aligned} \tag{70}$$

Again, a discussion similar to that after (28) can be made on each iterm of $H(\theta_h,\mathbf{k})$ as well as on the role of the size t. With $H_t(\theta_h,\mathbf{k})$ given by (70), we can also recursively implement parameter learning (58) that pushes $q(x|y)$ and $q(y)$ toward the structures of minimum complexity. Moreover, we can select a best k^* by (54).

F. A General Procedure for Parameter Learning

We further introduce a general procedure for parameter learning (53) with all the three typical architectures. Before doing so, we discuss some essential computational issues.

- **Computing $\max_y[q(x_t|y,\xi_t)q(y|\xi_t)]$:** In a B-architecture, at each x_t we need to get $y_t = F(x_t|\xi_t)$ by (67) for which we have to compute $\max_y[q(\overline{x}_t|y,\xi_t)q(y|\xi_t)]$. When $q(x_t|y,\xi_t)$ and $q(y|\xi_t)$ are both Gaussians with linear regression functions, it is a typical quadratic optimization that can be analytically solved from its linear critical equation

$$\nabla_y \ln[q(x_t|y,\xi_t)q(y|\xi_t)] = 0. \tag{71}$$

In other cases, (71) is usually a complicated nonlinear equation with many solutions, and thus finding $\max_y[q(x_t|y,\xi_t)q(y|\xi_t)]$ is a difficult task. Referred to Table II for details, we can solve this problem via three types of approximations:

1) *Real approximation:* When $q(x_t|y,\xi_t)$ is Gaussian with a linear regression between x_t and y, ξ_t, and $q(y|\xi_t)$ is Bernoulli, (71) is linear but y is binary. In this case, we can approximately solve y by regarding it as real and then turn it into binary via a threshold.
2) *Gaussian approximation:* When $q(x_t|y,\xi_t)$ is Gaussian with a linear regression between x_t and y, ξ_t, but $q(y|\xi_t)$ is non-Gaussian, we can approximate $q(y|\xi_t)$ by a Gaussian with a linear regression between y and ξ_t. Then, we solve (71).
3) *Fixed posteriori approximation:* When one or both of $q(x_t|y,\xi_t)$ and $q(y|\xi_t)$ are described by a mixture of Gaussians or Bernoulli densities, we consider $\nabla_u \ln[\sum_r \alpha(r|v)q_r(u|v)] \approx \sum_r \kappa^{(r)}(u|v)\nabla_u q_r(u|v)$, by approximately regarding that $\kappa^{(r)}(u|v) = \alpha_r q_r(u|v)/\sum_j \alpha_j q_j(u|v)$ is irrelevant to u.

- **Gradients $\nabla_{\theta_{x|y}} H_t(\theta,\mathbf{k})$ and $\nabla_{\theta_y} H_t(\theta,\mathbf{k})$:** We have

$$\begin{aligned}
\nabla_{\theta_{x|y}} H_t(\theta,\mathbf{k}) &= \sum_{\tau=1}^{t}\left(\frac{1}{t}-\gamma_t^b\right)\nabla_{\theta_{x|y}}\ln q(\overline{x}_t|\hat{y}_t,\hat{\xi}_t) \\
\gamma_t^b &= \frac{q(\hat{y}_t|\hat{\xi}_t)q(\overline{x}_t|\hat{y}_t,\hat{\xi}_t)}{z_q(t)} \\
\nabla_{\theta_y} H_t(\theta,\mathbf{k}) &= \sum_{\tau=1}^{t}\left(\frac{1}{t}-\gamma_t^f\right)\nabla_{\theta_y}\ln q(\hat{y}_\tau|\hat{\xi}_\tau) \\
\gamma_t &= \begin{cases}\gamma_t^b, & \text{for (65)},\\ \gamma_t^f, & \text{for (66)}\end{cases} \\
\gamma_t^f &= \frac{q(\hat{y}_t|\hat{\xi}_t)}{z_q^y(t)}.
\end{aligned} \tag{72}$$

For an online updating, we can simply consider $\nabla_{\theta_{x|y}}\ln q(\overline{x}_t|\hat{y}_t,\hat{\xi}_t)$ and $\nabla_{\theta_y}\ln q(\hat{y}_\tau|\hat{\xi}_\tau)$. Also, we can incrementally get $z_q(t) = z_q(t-1) + q(\hat{y}_t|\hat{\xi}_t)q(\overline{x}_t|\hat{y}_t,\hat{\xi}_t)$ and $z_q^y(t) = z_q^y(t-1) + q(\hat{y}_t|\hat{\xi}_t)$.

- **Computing Gradients via Chain Rule:** Given $\ln q(x_t|y,\xi_t)$ and $\ln q(y|\xi_t)$ with $y = f^\circ(Wx_t + H\xi_t + \mu)$, in help of the chain rule we get

$$\begin{aligned}
\nabla_W \ln q(\overline{x}_t|y,\hat{\xi}_t) &= D_f(y)\psi(y)\overline{x}_t^T \\
\nabla_w \ln q(\overline{x}_t|y,\hat{\xi}_t) &= D_f(y)\psi(y) \\
\nabla_W \ln q(y|\hat{\xi}_t) &= D_f(y)\phi(y)\overline{x}_t^T \\
\nabla_w \ln q(y|\hat{\xi}_t) &= D_f(y)\phi(y) \\
\psi(y) &= \nabla_y \ln q(\overline{x}_t|y,\hat{\xi}_t) \\
\phi(y) &= \nabla_y \ln q(y|\hat{\xi}_t) \\
D_f(y) &= \mathrm{diag}[f'(y^{(1)}),\cdots,f'(y^{(k)})] \\
f'(r) &= \frac{df(r)}{dr} \\
\nabla_{x_t}\ln q(x_t|\hat{y}_t,\hat{\xi}_t) &= \left[\frac{\partial\ln q(x_t|\hat{y}_t,\xi_t)}{\partial x_t} + W^T D_f(\hat{y}_t)\psi(\hat{y}_t)\right] \\
\nabla_{x_t}\ln q(y_t|\hat{\xi}_t) &= W^T D_f(y_t)\phi(y_t).
\end{aligned} \tag{73}$$

- **Lagrange Approach for Ascend Directions:** In a F-architecture, we need to get the ascend directions of $H(\theta,\mathbf{k})$ given in (66) subject to the constraints of $\Sigma_{yj} = D_j$ for $j = 1,\cdots,k_f$. When $f^\circ(y) = y$ and thus $y_{tj} = W_j x_t + K_j\xi_\tau + \mu_j$, it follows from (66) that $\Sigma_{yj} = W_j\Sigma_{xj}W_j^T$ with $\Sigma_{xj} = E[\gamma(j|x_t,\xi_t)(x_t-\mu_j)(x_t-\mu_j)^T]$. In this case, the constraint $\Sigma_{yj} = D_j$ only affect W_j. Thus, we consider the gradient of the Lagrange cost with respect to W_j, i.e., $\nabla_{W_j}H_L = \nabla_{W_j}\{H(\theta,\mathbf{k}) + \mathrm{Tr}[\Lambda_L(\Sigma_{yj}-D_j)]\} = \nabla_{W_j}H(\theta,\mathbf{k}) + M_W$, where Λ_L is a diagonal matrix consisting of Lagrange coefficients, and $M_W = \nabla_{W_j}\mathrm{Tr}[\Lambda_L(\Sigma_{yj}-D_j)] = 2\Lambda_L W_j\Sigma_{xj}$. Following [3], we further consider the natural gradient of H_L with respect to W_j, which is given by $(\nabla_{W_j}H_L)W_j^TW_j = \nabla_{W_j}H(\theta,\mathbf{k})W_j^TW_j + M_W W_j^TW_j$. Moreover, $M_W W_j^TW_j = 2\Lambda[W_j\Sigma_{xj}W_j^T]W_j = 2\Lambda_L D_j W_j$. Furthermore, $\nabla_{W_j}H(\theta,\mathbf{k}) = \gamma_t^f\phi(\hat{y}_t)\overline{x}_t^T$, $\phi(\hat{y}_t) = \nabla_y H(\theta,\mathbf{k})$ and $\gamma_t^f = (1/t) - (q(\hat{y}_t|\hat{\xi}_t)/z_q^y(t))$.

TABLE II
FINDING PEAK $\overline{v} = \arg\max_v \ln[p(u|v)p(v)]$ VIA SOLVING $\nabla_v \ln[p(u|v)p(v)] = 0$

(A) $p(u|v) = G(u|Av + c, \Sigma)$ and $p(v) = G(v|d, \Lambda)$

Solution: $\bar{v} = A_v^{-1}(u_A + \Lambda^{-1}d)$, (A)

$u_A = A^T\Sigma^{-1}(u - c), \quad A_v = A^T\Sigma^{-1}A + \Lambda^{-1}.$

(B) $p(u|v) = G(u|Av + c, \Sigma)$, $p(v) = \prod_j p_j^{v^{(j)}}(1 - p_j)^{1-v^{(j)}}$

(a) Get an exact solution via either enumerating y or a quadratic programming;

(b) Get an approximate solution via solving $\nabla_y \ln[G(u|Av + c, \Sigma)p(v)] = 0$ by

$$v_* = (A^T\Sigma^{-1}A)^{-1}\{A^T\Sigma^{-1}(x - c) + [d_1, \cdots, d_k]^T\},$$

$$\hat{v}^{(j)} = \begin{cases} 1, & \text{if } v_*^{(j)} > 0.5, \\ 0, & \text{otherwise;} \end{cases} \qquad d_j = \ln \tfrac{p_j}{1-p_j}.$$

(c) Extension to $p(v) = \sum_r \gamma_r \prod_j (p_{j,r})^{v^{(j)}}(1 - p_{j,r})^{1-v^{(j)}}$,

Get the above $\hat{v}$ with $d^{(j)}$ replaced by $\sum_r \eta^{(r)} \ln \frac{p_{j,r}}{1-p_{j,r}}$,

where $\sum_r \gamma_r = 1$, $\gamma_r > 0$, $\eta^{(r)} = \frac{\gamma_r \prod_j p_{j,r}^{v^{(j)}}(1-p_{j,r})^{1-v^{(j)}}}{p(v)}$.

(C) $p(u|v) = G(u|Av + c, \Sigma)$, $p(v) = \prod_j \sum_r \alpha_{j,r} G(v^{(j)}|m_{j,r}, \sigma_{j,r}^2)$

(a) Get a solution via Item (A) with $p(v)$ approximated by

a Gaussian $G(v|d, \Lambda)$, $d = [d_1, \cdots, d_k]^T$, $\Lambda = diag[\lambda_1, \cdots, \lambda_k]$,

$d_j = \sum_r \alpha_{j,r} m_{j,r}$, $\lambda_j = \sum_r \alpha_{j,r}[\sigma_{j,r}^2 + (m_{j,r} - d_j)^2]$.

(b) Get a solution via solving $\nabla_v \ln[G(u|Av + c, \Sigma)p(v)] = 0$

by regarding that $h_{j,r} = \frac{\alpha_{j,r} G(v^{(j)}|m_{j,r}, \sigma_{j,r}^2)}{p(v^{(j)})}$ is constant to v, resulting in

$$\hat{v} = (A^T\Sigma^{-1}A + diag[b_1, \cdots, b_k])^{-1}[A^T\Sigma^{-1}(u - c) + d],$$

$b^{(j)} = \sum_r \frac{h_{j,r}\sigma^2}{\sigma_r^2(j)}$, $d^{(j)} = \sum_r \frac{h_{j,r}\sigma^2 m_{j,r}}{\sigma_{j,r}^2}$.

(D) Extension to $p(u|v, \xi), p(v|\xi)$ witn ξ in consideration

(a) simply with $c, m_{j,r}$ in $p(u|v)$ replaced respectively by

$$B\xi + c, \quad m_{j,r} = e_{j,r}\xi_{j,r} + d_{j,r},$$

(b) with $d, p_j, p_{j,r}$ in $p(v)$ replaced respectively by

$$E\xi + d, p_j = s(e_j\xi_j + d_j), p_{j,r} = s(e_{j,r}\xi_{j,r} + d_{j,r}).$$

Note: $diag[d_1, \cdots, d_k]$ denotes a diagonal matrix consisting of $d_1, \cdots, d_k$.

As a result, the natural gradient direction of the Lagrange cost H_L with respect to W_j is given by

$$\begin{aligned}(\nabla_{W_j} H_L)W_j^T W_j &= \gamma_t^f \phi(\hat{y}_t)(W_j\overline{x}_t)^T W_j + 2\Lambda_L D_j W_j \\ &= \gamma_t^f[\phi(\hat{y}_t)(W_j\overline{x}_t)^T + \Lambda]W_j \\ \Lambda &= 2\Lambda_L D_j/\gamma_t^f \end{aligned} \tag{74}$$

which can be used as the ascend direction δW_j in (58) with $H(\theta, \mathbf{k})$ replaced by $H_t(\theta, \mathbf{k})$. From (66), D_j can be any fixed diagonal matrix. Thus, we can simply choose Λ to be any fixed diagonal matrix at our convenience, without worrying what value of Λ_L. For simplicity, we let $\Lambda = I$.

With the above preparation, we are ready to provide a general recursive procedure for parameter learning, with three typical architectures covered. Specifically, given $\hat{\xi}_t = [\hat{y}_{t-1}, \cdots, \hat{y}_{t-m}]^T$, the structure of $q(y_t|\xi_t)$ with its parameter set θ_y is given in Table I (where y, ξ are replaced by y_t and ξ_t, respectively), the structure of $q(y_t|\xi_t)$ with its parameter set $\theta_{x|y}$ is given by (60) and (61), and the structure of $p(y_t|x_t, \xi_t)$ with its parameter set $\theta_{y|x}$ is given by (62) and (63).

A General Recursive Procedure For Parameter Learning

Step 1) Get $\hat{y}_t$by (67).

Step 2)

$$\begin{aligned} z_q(t) &= z_q(t-1) + q(\hat{y}_t|\hat{\xi}_t)q(\overline{x}_t|\hat{y}_t, \hat{\xi}_t) \\ z_q^y(t) &= z_q^y(t-1) + q(\hat{y}_t|\hat{\xi}_t) \\ \gamma_t^b &= \frac{1}{t} - (1 - i_R)\frac{q(\hat{y}_t|\xi_t)q(\overline{x}_t|\hat{y}_t, \xi_t)}{z_q(t)} \\ \gamma_t^f &= \frac{1}{t} - (1 - i_R)\frac{q(\hat{y}_t|\xi_t)}{z_q^y(t)}. \end{aligned}$$

Step 3) From (72) we update

$$\begin{aligned}\theta_y^{new} = \theta_y^{old} + \zeta_y\gamma_t^f \nabla_{\theta_y}\{&\ln q(\hat{y}_t|\hat{\xi}_t) \\ &+ 0.5h_y^2 i_R \cdot \mathrm{Tr}[H_q(\hat{y}_t|\hat{\xi}_t)]\}.\end{aligned}$$

Step 4) Skip this step for a F-architecture, otherwise from (72) we update

$$\begin{aligned}\theta_{x|y}^{new} = \theta_{x|y}^{old} + \zeta_{x|y}\gamma_t^b \nabla_{\theta_{x|y}}\{&\ln q(\overline{x}_t|\hat{y}_t, \hat{\xi}_t) \\ &+ 0.5i_R \cdot \mathrm{Tr}[h_x^2 H_q^x(\overline{x}_t|\hat{y}_t, \hat{\xi}_t) + h_y^2 H_q^y(\overline{x}_t|\hat{y}_t, \hat{\xi}_t)]\}.\end{aligned}$$

Step 5) Skip this step for a B-architecture, otherwise, with j_* by (67) and $\phi(y)$, $\psi(y)$, $D_f(y)$ by (73)

(a) For a BI-architecture, with

$$y_t^e = D_f(\hat{y}_t)[\phi(\hat{y}_t) + \psi(\hat{y}_t) + i_R \Pi_H(\hat{y}_t)]$$

we update

$$\begin{aligned} W_{j_*}^{new} &= W_{j_*}^{old} + \zeta_{y|x}\gamma_t^b y_t^e \overline{x}_t^T \\ K_{j_*}^{new} &= K_{j_*}^{old} + \zeta_{y|x}\gamma_t^b y_t^e \hat{\xi}_t^T \\ \mu_{j_*}^{new} &= \mu_{j_*}^{old} + \zeta_{y|x}\gamma_t^b y_t^e; \end{aligned}$$

(b) For a F-architecture with $f^{\circ}(y) = y$ and thus $D_f(\hat{y}_t) = I$, from (74) with $\Lambda = I$ we update

$$\begin{aligned} W_{j_*}^{new} &= W_{j_*}^{old} + \zeta_y \gamma_t^f [I + \phi(\hat{y}_t)(W_{j_*}\overline{x}_t)^T] W_{j_*}^{old} \\ K_{j_*}^{new} &= K_{j_*}^{old} + \zeta_y \gamma_t^f \phi(\hat{y}_t)\hat{\xi}_t^T \\ \mu_{j_*}^{new} &= \mu_{j_*}^{old} + \zeta_y \gamma_t^f \phi(\hat{y}_t) \end{aligned} \tag{75}$$

where $\zeta_{x|y} > 0$, $\zeta_{y|x} > 0$ and $\zeta_y > 0$ are prefixed step sizes for updating, and γ_t^b, γ_t^f are dynamic step sizes that change as t for controlling delearning in implementing by normalization learning. γ_t^b is used in a BI-architecture or a B-architecture, while γ_t^f is used in a F-architecture. i_R is a prespecified indicator that takes either $i_R = 0$ for implementing normalization learning or $i_R = 1$ for implementing data smoothing learning. Specifically, we have three major situations:

- When $i_R = 0$, the normalization learning is implemented based on (65) or (66). Specifically, Step 3 and Step 4 are featured by the updating form $\theta^{new} = \theta^{old} + \eta\delta\theta$ that ascends $\ln p(u|v,\theta)$, with the detailed $\delta\theta$ in several typical cases given in Table III.
- When $i_R = 1$, the data smoothing learning (47) is implemented by (75) with h_x, h_y fixed. We have two choices for updating h_x, h_y. One is simulated annealing [38] via letting h_x, h_y to start at initial values large enough and then gradually to reduce to zero during the implementation of (75). The other is to alternatively make the implementation of (75) with h_x, h_y fixed and make the following updating with θ fixed:

$$h_x^{new} = h_x^{old} + \eta_h \left\{ \frac{d_x}{h_x} - \frac{d\ln z_q^t(h,k)}{dh_x} + h_x \,\mathrm{Tr}[H_q^x(\overline{x}_t|\hat{y}_t,\hat{\xi}_t)] \right\}, \text{ or}$$

$$h_x^{\text{new2}} \approx \frac{2h_{x,0}^2}{1+\sqrt{1+4h_{x,0}^2 d_x^{-1}\gamma_x}}$$

$$\lambda_x = \frac{1}{t}\sum_{\tau=1}^{t} \mathrm{Tr}[H_q^x(\overline{x}_t|\hat{y}_t,\hat{\eta}_t)]$$

$$\begin{aligned} h_y^{new} = h_y^{old} + \eta_h \Big\{ &\frac{k}{h_y} - \frac{d\ln z_q^t(h,k)}{dh_y} \\ &+ h_y \,\mathrm{Tr}[H_q^y(\overline{x}_t|\hat{y}_t,\hat{\xi}_t) + H_q(\hat{y}_t|\hat{\xi}_t)] \Big\}, \quad \text{or} \end{aligned}$$

$$h_y^{\text{new2}} \approx \frac{2h_{y,0}^2}{1+\sqrt{1+4h_{y,0}^2 k - 1\lambda_y}}$$

$$\lambda_y = \frac{1}{t}\sum_{\tau=1}^{t} \mathrm{Tr}[H_q^y(\overline{x}_t|\hat{y}_t,\hat{\xi}_t) + H_q(\hat{y}_t|\hat{\xi}_t)] \tag{76}$$

$$h_{x,0}^2 = \frac{1}{d_x}\sum_{\tau=1}^{N}\sum_{t=1}^{N} \gamma_{t,\tau}||x_t - x_\tau||^2,$$

$$h_{y,0}^2 = \frac{1}{k}\sum_{\tau=1}^{N}\sum_{t=1}^{N} \gamma_{t,\tau}||y_t - y_\tau||^2, \gamma_{t,\tau} = \frac{e^{-0.5||x_t - x_\tau||^2 + ||y_t - y_\tau||^2}}{z_q(h^{\text{old2}},k)}$$

which is obtained from Step 2) in (34).

Moreover, we also have $\Pi_H(y_t)$ in Step 5(a) given by

$$\begin{aligned} \Pi_H(y_t) = \nabla_y \{ &0.5h_x^2 \,\mathrm{Tr}[H_q^x(\overline{x}_t|y_t,\hat{\xi}_t)] \\ &+ 0.5h_y^2 \,\mathrm{Tr}[H_q^y(\overline{x}_t|y_t,\hat{\xi}_t) + H_q(y_t|\hat{\xi}_t)] \} \end{aligned} \tag{77}$$

which becomes zero when $\mathrm{Tr}[H_q^x(\overline{x}_t|y_t,\hat{\xi}_t)]$, $\mathrm{Tr}[H_q^y(\overline{x}_t|y_t,\hat{\xi}_t)]$ and $\mathrm{Tr}[H_q(y_t|\hat{\xi}_t)]$ are irrelevant to y, e.g., they are the constant covariance matrices when $q(x_t|y_t,\xi_t)$ and $q(y_t|\xi_t)$ are Gaussians.

- Empirical learning is implemented by (75) when either $i_R = 0$ with $\gamma_t^b = 1$, $\gamma_t^f = 1$ or $i_R = 1$ with $h_x = 0$, $h_y = 0$.

Furthermore, the above temporal learning procedure directly applies to the nontemporal case by simply setting $\xi_t = 0$ and discarding all the updating equations for parameters that correspond to ξ_t. In Step 4, e.g., for $q(x_t|y_t,\xi_t)$ by (60) and (61), we can ignore those updating rules on B_j by simply setting $B_j = 0$. Also, in Step 5 we can ignore those updating rules on K_j by simply setting $K_j = 0$.

III. BYY Independent State Space and Generalized APT Analyses

With $q(y_t|\xi_t)$ given by Table I, which covers the cases of (9), (12), and (14), the BYY independent state space system in help of the general adaptive parameter learning procedure (75) provides a general tool for implementing a number of generalized APT analyses. In this section, we introduce several typical examples. For simplicity, we focus on modeling the first-order serial relation, i.e., $\xi_t = y_{t-1}$. However, all the results and discussions can be directly extended to the cases of modeling a higher order serial relation, simply by using ξ_t to replace y_{t-1}.

A. B-Architecture: Temporal Factor analyses and Generalized APT analyses

- **Factor Analysis Versus Temporal Factor Analysis:** We further make a detailed consideration on the B-architecture part in (75), starting at the classic state-space model (13) with a diagonal B as well as Gaussains $G(\varepsilon_t|0,\Lambda)$ and $G(e_t|0,\Sigma)$. We call this special case Gaussian temporal factor analysis (TFA) since it returns to Gaussian factor analysis when $B = 0$. As shown in [63], one advantage of Gaussian TFA is that the temporal relation in (13) removes out the rotation indeterminacy (6) that is suffered by the Gaussian factor analysis.

In this case, (75) is simplified into the following algorithm:

Step 1)

$$\begin{aligned} \hat{y}_t &= [\Lambda^{-1} + A^T\Sigma^{-1}A]^{-1}(A^T\Sigma^{-1}\overline{x}_t + \Lambda^{-1}B\hat{y}_{t-1}) \\ \varepsilon_t &= \hat{y}_t - B\hat{y}_{t-1}, \quad e_t = \overline{x}_t - A\hat{y}_t. \end{aligned}$$

TABLE III
ADAPTIVE RULES FOR UPDATING $\theta^{new} = \theta^{old} + \eta\delta\theta$ TO ASCEND $\ln p(u|v,\theta)$

(A) Gaussian: $p(u|v,\theta) = G(u|Av+c,\Sigma)$, $y = Av$,

$$e = u - c,\ \delta c = e,\ \delta A = ev^T,\ \delta\Sigma = \begin{cases} diag[ee^T - \Sigma], & \Sigma \text{ is diagonal,} \\ ee^T - \Sigma, & \text{otherwise.} \end{cases}$$

(B) Bernoulli: $p(u|v,\theta) = \prod_j s(y^{(j)})^{u^{(j)}}(1 - s(y^{(j)}))^{1-u^{(j)}}$

$$s(r) = 1/(1+\epsilon^{-r}),\ y = Bv + c,\ \epsilon^{(j)} = u^{(j)} - s(y^{(j)}),\ \delta c = e,\ \delta B = ev^T.$$

(C) Independent mixture : $p(u|v,\theta) = \prod_j \sum_{r=1}^{\kappa_j} \beta_{j,r}(v^{(j)}) p(u^{(j)}|v^{(j)},\theta_{j,r})$,

$$\beta_{j,r}(v^{(j)}) = e^{z_{j,r}} / \sum_{r=1}^{\kappa_j} e^{z_{j,r}},\quad z_{j,r} = b_{j,r}v^{(j)} + c_{j,r}$$

$$p(u^{(j)}|v^{(j)},\theta_{j,r}) = \begin{cases} G(u^{(j)}|y_r^{(j)},\sigma_{j,r}^2), & \text{Gaussian,} \\ s(y_r^{(j)})^{u^{(j)}}(1 - s(y_r^{(j)}))^{1-u^{(j)}}, & \text{Bernoulli,} \end{cases}$$

$$y_r^{(j)} = a_{j,r}v^{(j)} + m_{j,r},\ \text{update } h_{j,r}(v^{(j)}) = \frac{\beta_{j,r}(v^{(j)}) p(u^{(j)}|v^{(j)},\theta_{j,r})}{\sum_{r=1}^{\kappa_j} \beta_{j,r}(v^{(j)}) p(u^{(j)}|v^{(j)},\theta_{j,r})}.$$

$$\epsilon_{j,r} = \sum_{i=1}^{\kappa_j} h_{j,i}(v^{(j)})[I_{i,r} - \beta_{j,r}(v^{(j)})],$$

$$\delta c_{j,r} = e_{j,r},\ \delta b_{j,r} = e_{j,r}v^{(j)},$$

$$I_{i,r} = \begin{cases} 1, & i = r, \\ 0, & \text{otherwise;} \end{cases} \qquad e_{j,r} = \begin{cases} u^{(j)} - y_r^{(j)}, & \text{Gaussian,} \\ u^{(j)} - s(y_r^{(j)}), & \text{Bernoulli.} \end{cases}$$

$$\delta m_{j,r} = h_{j,r}(v^{(j)})e_{j,r},\ \delta a_{j,r} = h_{j,r}(v^{(j)})e_{j,r}v^{(j)},\ \delta\sigma_{j,r}^2 = h_{j,r}(v^{(j)})(\epsilon_{j,r}^2 - \sigma_{j,r}^2),$$

$$\text{Particularly, } \delta a_j = \sum_{r=1}^{\kappa_j} \delta a_{j,r} \text{ when } a_{j,r} = a_j, b_{j,r} = 0. \tag{84}$$

(D) Mixture-of-Experts: $p(u|v,\theta) = \sum_r \alpha_r(v) p(u|v,\theta_r)$

$$\alpha_r(v) = \begin{cases} e^{z^{(r)}} / \sum_j e^{z^{(j)}},\ z^{(j)} = w_j^T v + d_j, & \text{Soft-max gate,} \\ \frac{\beta_r G(v|m_r,\Pi_r)}{\sum_j \beta_j G(v|m_j,\Pi_j)},\ \beta_r = \frac{e^{d^{(r)}}}{\sum_j e^{d^{(j)}}}, & \text{Gaussian gate;} \end{cases}$$

$$p(u|v,\theta_r) = \begin{cases} G(u|A_r v + c_r, \Sigma_r), & \text{Gaussian,} \\ \prod_{j=1}^{k} s(y_r^{(j)})^{u^{(j)}}(1 - s(y_r^{(j)}))^{1-u^{(j)}}, & \text{Bernoulli;} \end{cases}$$

$$y_r = A_r v + c_r.$$

(i) Updating the Soft-max gate: $h_r(v) = \frac{\alpha_r(v) p(u|v,\theta_r)}{p(u|v,\theta)}$,

$$e_r = \sum_i h_i(v)[I_{i,r} - \alpha_r(v)],\ \delta d_r = e_r,\ \delta w_r = e_r v;$$

(ii) Updating the Gaussian gate: $h_r(v) = \frac{\beta_r G(v|m_r,\Pi_r) p(u|v,\theta_r)}{\sum_j \beta_j G(v|m_j,\Pi_j) p(u|v,\theta_j)}$,

$$\delta\beta^{(r)} = \sum_i h_i(v)(I_{i,r} - \beta_r),\ e_r^g = v - m_r,$$

$$\delta m_r = h_r(v)e_r^g,\ \delta\Pi_r = h^{(r)}(u)[e_r^g e_r^{g\,T} - \Pi_r].$$

(iii) Updating Gaussian Experts: $\epsilon_r = u - c_r - A_r v$, $\delta c = h_r(v)e_r$,

$$\delta A_r = h_r(v)e_r v^T,\ \delta\Sigma_r = h_r(u)(e_r e_r^T - \Sigma_r).$$

(iv) Updating Bernoulli Experts: $e_r^{(j)} = v^{(j)} - s(y_r^{(j)})$,

$$\delta c_r = e_r,\ \delta A_r = h_r(v)e_r v^T.$$

Note: diag[M] is a diagonal matrix that takes the diagonal part of a matrix M.

Step 2)

$$\gamma_t^b = \frac{1}{t} - (1 - i_R)\frac{G(\varepsilon_t|0,\Lambda^{old})G(e_t|0,\Sigma^{old})}{z_q(t)}$$

$$z_q(t) = z_q(t-1) + G(\varepsilon_t|0,\Lambda^{old})G(e_t|0,\Sigma^{old}).$$

Step 3)

$$B^{new} = B^{old} + \eta\gamma_t^b\ \text{diag}[\varepsilon_t\hat{y}_{t-1}^T]$$

$$\Lambda^{new} = (1 - \eta\gamma_t^b)\Lambda^{old} + \eta\gamma_t^b\ \text{diag}[\varepsilon_t\varepsilon_t^T + i_R h_y^2 I].$$

Step 4)

$$A^{new} = A^{old} + \eta\gamma_t^b[e_t\hat{y}_t^T - i_R h_y^2 A^{old}]$$

$$\Sigma^{new} = (1 - \eta\gamma_t^b)\Sigma^{old} + \eta\gamma_t^b[e_t e_t^T + i_R(h_x^2 I + h_y^2 AA^T)]. \tag{78}$$

Here, Steps 1–4 are the specific forms of Steps 1–4 in (75), respectively, with Item (A) in Table II for Step 1 and Item (A) in Table III for Step 3 and Step 4. In the sequel, we look several typical cases:

a) When $i_R = 1$, $h_y = 0$, $h_x = 0$, $\Lambda = I$, (78) is simplified into

$$\hat{y}_t = [I + A^T\Sigma^{-1}A]^{-1}(A^T\Sigma^{-1}\overline{x}_t + B\hat{y}_{t-1})$$

$$\varepsilon_t = \hat{y}_t - B\hat{y}_{t-1},\quad e_t = \overline{x}_t - A\hat{y}_t$$

$$B^{new} = B^{old} + \eta\ \text{diag}[\varepsilon_t\hat{y}_{t-1}^T],\quad A^{new} = A^{old} + \eta e_t\hat{y}_t^T$$

$$\Sigma^{new} = (1 - \eta)\Sigma^{old} + \eta e_t e_t^T. \tag{79}$$

At the special case $B = 0$ it is actually a variant of the adaptive algorithm for implementing Gaussian FA previously given [68, Sec. 4.2.4]. When B is not forced to be zero, (79) acts as a new variant of the adaptive algorithm previously given by Table III in [63] for Gaussian TFA.

b) Equation (79) is further extended by relaxing Λ from I to a diagonal matrix that is determined via learning

$$\Lambda^{new} = (1-\eta)\Lambda^{old} + \eta \,\mathrm{diag}[\varepsilon_t \varepsilon_t^T]. \tag{80}$$

As a result, for both Gaussian FA and Gaussian TFA, automated model selection becomes possible via pushing some of diagonal elements to zero, when k is large enough. In implementation, we can simply remove a diagonal element of Λ if it becomes zero since it corresponds either a constant or a simple deterministic exponential decaying series, both of which are not much useful. Then, we continue to run the algorithm until it converges with no diagonal element of Λ still tending to zero.

c) When $i_R = 1$ with $h_y > 0$, $h_x > 0$, learning by (78) implements data smoothing that regularizes the effect of finite number of samples. Again, we can update h_x, h_y via either simulated annealing or (76) which is now simplified with

$$\lambda_z = \mathrm{Tr}[\Sigma^{-1}], \quad \lambda_y = \mathrm{Tr}[A^T\Sigma^{-1}A + \Lambda^{-1}]. \tag{81}$$

d) When $i_R = 0$, learning by (78) implements normalization learning with delearning in effect.

In the case that $h_y > 0$ or Λ is simply fixed at $\Lambda = I$, though the least complexity nature will be in effect during learning by (78), the scale k is prevented to be further reduced. In such cases, we can enumerate k incrementally, and then select an approriate scale k by (54), which is simplified as follows

$$\min_k J(k) = 0.5 \begin{cases} J_y(k) + \ln|\Sigma|, & \text{(a) empirical} \\ J_y(k) + \ln|\Sigma| + 2\ln z_q(t), & \text{(b) normalization} \\ J_y(k) + \ln|\Sigma| + h_y^2(k + \mathrm{Tr}[A^T\Sigma^{-1}A]) \\ \quad -k\ln(2\pi h_y^2), & \text{(c) data smoothing,} \end{cases}$$
$$J_y(k) = k\ln(2\pi) + k \tag{82}$$

which is obtained from (70) after discarding those irrelevant items. Also, from the relation between Gaussian factor analysis and PCA discussed in Section I, we can directly use (82) at the special case $\Sigma = \sigma^2 I$ for the subspace dimension with $\ln|\Sigma| = d\ln\sigma^2$, where σ^2 is given by the average of the $d-k$ least eigenvalues of the sample covariance matrix of x. The details are referred to [67].

- **Non-Gaussian FA Versus NonGaussian TFA:** Another direction that extends Gaussian TFA (78) is to consider a nonGaussian noise ε_t in the state equation $y_t = By_{t-1} + \varepsilon_t$. We consider, e.g., $q(y_t|y_{t-1},\theta_y)$ given by Item (C) in Table III with each $p(u^{(j)}|v^{(j)},\theta_{j,r})$ being Gaussian, u being y_t and v being y_{t-1}, which can be better understood from its following two special cases:
  a) *Non-Gaussian FA:* When $a_{j,r} = 0$, $b_{j,r} = 0$, for the density by Item (C) in Table III with Gaussian $p(u^{(j)}|v^{(j)},\theta_{j,r})$, we have $y_t = \varepsilon_t$ that is non-Gaussian and component-wise independent, with each component described by an one variable Gaussian mixture. That is, the temporal relation between y_t, y_{t-1} has been removed by setting $B = 0$. This degenerated case of non-Gaussian TFA is actually an extension of Gaussian factor analysis and thus we call it Non-Gaussian FA. In this case, we can get a considerably improvement over the previous nonGaussian FA algorithm given in [68, Eqs. (35), (37), and (38)].
  b) *Non-Gaussian TFA:* When $a_{j,r} = a_j$, $b_{j,r} = 0$, for the density by Item (C) in Table III with Gaussian $p(u^{(j)}|v^{(j)},\theta_{j,r})$, we have a linear additive model $y_t = By_{t-1} + \varepsilon_t$ with ε_t is non-Gaussian. In this case, we get non-Gaussian TFA.

Generally, the case of $q(y_t|y_{t-1},\theta_y)$ given by Item (C) in Table III describes a temporal relation between $y_t^{(j)}$, $y_{t-1}^{(j)}$ on each dimension in a piecewise way by a mixture of expert that weights a number of linear relations via a gate. We call this general case *Generalized Non-Gaussian TFA*, which is implemented via modifying (78) according to (75). Also, (82) is correspondingly modified with $J_y(k)$ replaced by

$$J_y(k) = -\frac{1}{N}\sum_{t=1}^{N} \ln q(\hat{y}_t|\hat{y}_{t-1},\theta_y). \tag{83}$$

- **Non-Gaussian Observation Noise and Nonlinear Generative Model:** Extensions can be further made with $x_t = Ay_t + e_t$ in (13) replaced by a mixture-of-experts density $p(x_t|y_t,\theta_{x|y})$ in Item (D) of Table III. For a Gaussian expert $G(u|A_r v + c_r, \Sigma_r)$, when $A_r = A$, $w_r = 0$ for all r, $p(x_t|y_t,\theta_{x|y})$ describes an additive model $x_t = Ay_t + e_t$, where non-Gaussian noise e_t is modeled by a mixture of Gaussians with different mean c_r and Σ_r. While $p(x_t|y_t,\theta_{x|y})$ describes a nonlinear model in a piecewise linear way under a Gaussian noise e_t, when $\Sigma_r = \Sigma$ for all r.
- **Binary Factor Analysis Versus Independent HMM:** Another non-Gaussian extension is to consider $q(y_t|y_{t-1},\theta_y)$ by Item (C) in Table III with $p(u^{(j)}|v^{(j)},\theta_{j,r})$ being *Bernoulli*, with u being y_t and v being y_{t-1}. This case can be better understood from the following two special cases:
  a) *Bernoulli FA:* For $\kappa_j = 1$, $q(y_t|y_{t-1},\theta_y)$ reduces into a *Bernoulli* density when $a_{j,1} = 0$, $b_{j,1} = 0$. In this case, $x_t = Ay_t + e_t$ is generated from independent *Bernoulli* binary factors, and thus is called *Bernoulli factor analysis*. There is no need to consider the cases of $\kappa_j > 1$, a finite mixture of k *Bernoulli* densities is still a *Bernoulli* density since when $a_{j,1} = 0$, $b_{j,1} = 0$.
  b) *Independent HMM:* The temporal relation between y_t, y_{t-1} is taken in consideration by $q(y_t|y_{t-1},\theta_y)$ when $a_{j,1} \neq 0$. In this case, the series y_t, y_{t-1}, $\cdots$ consists of k independent Markov chains which are hidden to the observed series x_t, x_{t-1}, $\cdots$. That is, we actually encounter an independent hidden Markov model (HMM). When $\kappa_j = 1$, the temporal relation between $y_t^{(j)}$, $y_{t-1}^{(j)}$ is set up via a postlinear regression $s(a_j y_{t-1}^{(j)} + m_{j,r})$. When $\kappa_j > 1$, the temporal relation between $y_t^{(j)}$, $y_{t-1}^{(j)}$ is set up via $\sum_{r=1}^{\kappa_j} \beta_{j,r}(v^{(j)}) s(a_{j,r} y_{t-1}^{(j)} + m_{j,r})$, which reduces to $\sum_{r=1}^{\kappa_j} \beta_{j,r} s(a_j y_{t-1}^{(j)} + m_{j,r})$ under the constraints $a_{j,r} = a_j$, $b_{j,r} = 0$.

Similar to (78), from (75) we can implement the *Bernoulli FA* and *Independent HMM* by

Step 1) Get $\hat{y}_t$ in help of Items (B) and (D) in Table II;

Step 2) $\gamma_t^b = \dfrac{1}{t} - (1 - i_R)\dfrac{q(\hat{y}_t|\hat{y}_{t-1})q(\overline{x}_t|\hat{y}_t, \hat{y}_{t-1})}{z_q(t)}$,
$z_q(t) = z_q(t-1) + q(\hat{y}_t|\hat{y}_{t-1})q(\overline{x}_t|\hat{y}_t, \hat{y}_{t-1})$;

Step 3) With the substitutions of u by y_t and v by y_{t-1}, as well as η by $\eta\gamma_t^b$, we have that

a) For Bernoulli FA, $q(y_t|y_{t-1}, \theta_y)$ is given and updated as in Item (B) of Table III;

b) For Independent HMM, $q(y_t|y_{t-1}, \theta_y)$ in a mixture of Bernoullis is given and updated as in Item (C) of Table III, [see (84)];

Step 4)

(a) For a real $x_t = Ay_t + e_t$ with $G(e_t|0, \Sigma)$, update $e_t = \overline{x}_t - A\hat{y}_t$, $A^{new} = A^{old} + \eta\gamma_t^b e_t \hat{y}_t^T$, $\Sigma^{new} = (1 - \eta\gamma_t^b)\Sigma^{old} + \eta\gamma_t^b[e_t e_t^T + i_R h_x^2 I]$,

(b) For a binary x_t, $p(x_t|y_t, \theta_{x|y})$ is given and updated as in either Item (B) or Item (C) of Table III, with the substitutions of u by x_t and v by y_t.

Specifically, the above algorithm is implemented via either normalization learning when $i_R = 0$ or data smoothing learning when $i_R = 1$. Particularly, both cases will reduce to empirical learning when either $i_R = 0$, $\gamma_t^b = 1$ or $i_R = 1$, $h_x = 0$.

The Bernoulli FA case is an improved variant of the adaptive algorithm firstly given in [68, p. 238, eq. (1)] for binary factor analysis. The independent HMM case with $\kappa_j = 1$ is an improved variant of the independent HMM algorithm previously given by [64, Tab. IV].

Since y_t is binary, $q(y_t|y_{t-1})$ consists of δ-densities. Thus in (84), $q(\hat{y}_t|y_{t-1})$ can be regarded as probability after divided by $\delta_y(0)$. In the case of $i_R = 0$ for normalization learning, the effect of this $\delta_y(0)$ is cancelled out by letting $z_q(t)$ also divided by $\delta_y(0)$. While in the case $i_R = 1$ for data smoothing learning, $h_y = 0$, which results in $-\ln h_y^k = \delta_y(0)$ that cancels out the effect of $\delta_y(0)$ from $q(\hat{y}_t|y_{t-1})$.

Again, model selection will be made automatically during the implementation of (84), when k is initialized large enough. During learning, some $s(a_{j,r}y_{t-1}^{(j)} + m_{j,r})$ will tend to either one or zero as either $a_{j,r}$ or $m_{j,r}$ diverges. In this case, we can simply remove the corresponding $a_{j,r}$, $m_{j,r}$, which effectively makes k reduced.

Alternatively, we can also enumerate k incrementally from a small value and running (84). Then, for the case of a real x_t, we select a best k^* by (82), where a Bernoulli $q(y_t|y_{t-1}, \theta_y)$ as in either Item (B) or Item (C) of Table III is used. By Item (B), e.g., $J_y(k)$ becomes

$$J_y(k) = -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k}[s(y_r^{(j)})\ln s(y_r^{(j)}) + (1 - s(y_r^{(j)})) \cdot \ln(1 - s(y_r^{(j)}))]. \quad (85)$$

While for a binary x_t with $q(x_t|y_t, y_{t-1})$ given by Item (B) in Table III, the term $0.5\ln|\Sigma|$ in (82) should be replaced by

$$-\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{d_x}[s(\hat{x}_t^{(j)})\ln s(\hat{x}_t^{(j)}) + (1 - s(\hat{x}_t^{(j)}))\ln(1 - s(\hat{x}_t^{(j)}))] \quad (86)$$

where $\hat{x}_t = Ay_t + c$ and

$$q(x_t|y_t, y_{t-1}) = \prod_j[s(\hat{x}_t^{(j)})\delta(1 - x_t^{(j)}) + (1 - s(\hat{x}_t^{(j)}))\delta(x_t^{(j)})].$$

- **Generalized APT Analyses:** In the existing statistical factor analysis for APT, the requirement on knowing A in (1) is exchanged by assuming that y_t is a standard Gaussian and uncorrelated (i.e., second-order independent in components). However, as discussed in Section I, some empirical tests [25], [1] failed to provide a strong support. Here, we still believe that it is on a right direction to assume that the security returns are affected by factors that mutually have a minimum dependence on each other. However, the use of only the second-order independence leads to the intrinsic indeterminacy (6). It is this indeterminacy and several problems discussed in Section I that affect the performances considerably. As above introduced, the indeterminacy (6) can be removed by considering either higher order independence (9) in help of non-Gaussian $q(y|\xi)$ or temporal relation between y_t and y_{t-1}.

Moreover, the above various models also provide tools for solving the other problems in Section I. Specifically, we are able to generalize the existing APT analysis from the following perspectives:

a) Being different from [18], the equation in Step 1 of (78) provides an alternative solution for the cross-sectional approach discussed in Section I for estimating y_t. The equation actually provides an optimal inverse of the APT model (1) in help of estimating *a priori* density $q(y|\xi) = q(y_t)$ when $B = 0$ as well as using the temporal relation when $B \neq 0$.

b) Using the adaptive algorithm (79) for implementation, with $G(\varepsilon_t|0, \Lambda)$ and $G(e_t|0, \Sigma)$ as well as a diagonal B, the state space model (13) provides a solution to the *Problem (a) and Problem (f)* in Section I, i.e., we get a temporal extension of the statistical APT by considering $y_t = By_{t-1} + \varepsilon_t$ such that the rotation indeterminacy (6) is removed due to temporal relation.

c) The parameter learning with automated model selection or making model selection via the criterion (82) provides a tool for deciding the number of factors, i.e., we get a solution to the *Problem (b)* in Section I [20].

d) The *non-Gaussian FA* with (78) for implementation also solves the rotation indeterminacy (6) of the *Problem (a)*, in help of considering higher order independence and temporal relation. Also, the *Problem (b)* can be solved by (83) in this case.

e) The *Binary Factor Analysis and Independent HMM* implemented by (84) provide solutions of the *Problem (e)*, which acts as a type of *independent binary APT model* that looks more appealing by noticing that news in capital market is usually binary (e.g., a particular news "come" or "not come").

f) The generalized APT analyses can also be made with non-Gaussian observation noise and nonlinear generative

model in consideration, as efforts to solve the *Problem (c)* and *Problem (d)*.

B. Temporal ICA, Competitive ICA, LMSER+ICA and Temporal Extensions

We further discuss typical cases of (75) on a F-architecture and a BI-architecture, which provide some alternative or simplified solutions for implementing APT analyses.

- **F-Architetcure: Gaussain TICA, Non-Gaussian TICA and Competitive ICA:** We start at a simple F-architecture with $p(y_t|x_t, y_{t-1}) = \delta(y_t - (Wx_t + Ky_{t-1} + w))$ and $q(y_t|y_{t-1}) = G(y_t|By_{t-1}, I)$. With $\phi(y_t) = -(y_t - By_{t-1})$ and a diagonal B, the algorithm (75), especially its Step 5(b), is simplified into the following steps.

Step 1)

$$\hat{y}_t = W\overline{x}_t + K\hat{y}_{t-1} + w, \quad \varepsilon_t = \hat{y}_t - B\hat{y}_{t-1}.$$

Step 2)

$$\begin{aligned} W^{new} &= W^{old} + \zeta_{y|x}[I - \varepsilon_t(W^{old}x_t)^T]W^{old} \\ w^{new} &= w^{old} - \zeta_{y|x}\varepsilon_t \\ K^{new} &= K^{old} - \zeta_{y|x}\varepsilon_t\hat{y}_{t-1}^T \\ B^{new} &= B^{old} + \zeta_{y|x}\,\mathrm{diag}[\varepsilon_t\hat{y}_{t-1}^T]. \end{aligned} \tag{87}$$

We call this algorithm *Gaussian temporal ICA*. As shown in [63], without the rotation indeterminacy (6) it is able to make $y_t = Wx_t + K\hat{y}_{t-1} + w$ become independent in components, because of considering the temporal relation between y_t and y_{t-1}.

Generally, we consider a F-architecture that consists of $p(y_t|x_t, \xi_t)$ given by (62) and (63) with $f^{\circ}(y) = y$ and $q(y_t|y_{t-1})$ given by Item (C) in Table III, and with u replaced by y_t and v by y_{t-1}. After discarding Step 4 and those items irrelevant to a F-architecture, from (75) we can get the following steps.

Step 1) By (67), get

$$\begin{aligned} \hat{y}_t &= W_{j_*}\overline{x}_t + K_{j_*}\hat{y}_{t-1} + \mu_{j_*} \\ j_* &= \arg\max_j q(\hat{y}_{tj}|\hat{y}_{t-1}), \\ \hat{y}_{tj} &= W_j\overline{x}_t + K_j\hat{y}_{t-1} + \mu_j. \end{aligned}$$

Step 2)

$$\begin{aligned} z_q^y(t) &= z_q^y(t-1) + q(\hat{y}_t|y_{t-1}), \\ \gamma_t^f &= \frac{1}{t} - \frac{q(\hat{y}_t|y_{t-1})}{z_q^y(t)}. \end{aligned}$$

Step 3) Update $\theta_y^{new} = \theta_y^{old} + \zeta_y\gamma_t^f\delta\theta_y$, with $\delta\theta_y$ given as in Item (C) of Table III.

Step 4)

$$\begin{aligned} W_{j_*}^{new} &= W_{j_*}^{old} + \zeta_{y|x}\gamma_t^f[I + \phi(\hat{y}_t)(W_{j_*}^{old}\overline{x}_t)^T]W_{j_*}^{old}. \\ K_{j_*}^{new} &= K_{j_*}^{old} + \zeta_{y|x}\gamma_t^f\phi(\hat{y}_t)\hat{y}_{t-1}^T. \\ \mu_{j_*}^{new} &= \mu_{j_*}^{old} + \zeta_{y|x}\gamma_t^f\phi(\hat{y}_t). \end{aligned} \tag{88}$$

which is implemented via normalization learning that reduces to empirical learning when $\gamma_t^f = 1$.

In the special case of $p(y_t|x_t, \xi_t)$ by (62) with $k_f = 1$, we have only one linear function $y_t = Wx_t + Ky_{t-1} + \mu$. In this case, we can drop the equation for j_* and every appearance of the index j_* in (88). Then, we can get further insights by considering three typical cases of $q(y_t|y_{t-1})$ as follows:

i) $q(y_t|y_{t-1})$ describes a linear model $y_t = By_{t-1} + \varepsilon_t$ with ε_t being non-Gaussian, when $a_{j,r} = a_j$, $b_{j,r} = 0$, and is given by Item (C) in Table III with Gaussian $p(u^{(j)}|v^{(j)}, \theta_{j,r})$. In this case, the above (88) is actually equivalent to the temporal ICA algorithm (TICA) given by [63, eq. (42)]. Moreover, it is not difficult to observe that this TICA reduces to the above Gaussian TICA (87) at $\kappa_j = 1$.

ii) $q(y_t|y_{t-1})$ describes a Gaussian mixture, if we further impose $a_{j,r} = 0$, $b_{j,r} = 0$, and is given by Item (C) in Table III with Gaussian $p(u^{(j)}|v^{(j)}, \theta_{j,r})$. Fixing $w = 0$, $K = 0$ and thus $y_t = Wx_t$, both the above TICA and (88) further degenerate to the previously proposed learned parametric mixture based ICA [69], [72], [79]that improves the ICA algorithm [3], [8], as discussed in Section I.

iii) Generally, given by Item (C) in Table III with Gaussian $p(u^{(j)}|v^{(j)}, \theta_{j,r})$, $q(y_t|y_{t-1})$ describes the temporal relation between $y_t^{(j)}$, $y_{t-1}^{(j)}$ in a piecewise way by a mixture of expert. We call (88) in this case *generalized TICA*.

Furthermore, for the cases of $p(y_t|x_t, \xi_t)$ in (62) with $k_f > 1$, (88) implements k_f different generalized TICAs via a WTA competition in Step 1. Each of these k_f generalized TICAs works locally on a segment of a time series at the location μ_j. Thus, (88) in such a case is called *competitive TICA* or particularly *competitive ICA* in the above case ii), which is more suitable for a series of $\mathbf{x} = \{x_t\}$ with a number of different local statistical properties.

- **BI-Architetcure: LMSER, LMSER+ICA and Temporal Extensions:** The B-architectures and the F-architectures both have advantages and weak points. A B-architecture focuses on how data is reconstructed by $q(x_t|y_t, y_{t-1})$ from factors of y_t to fit the observed data x_t, such that not only noise is taken in consideration but also the factors can be evaluated as principal or minor in a sense of the fitting goodness, which makes meaningful the problem of model selection for the best factors or structures of minimum complexities. However, the disadvantage of a B-architecture is its expensive computing cost both on learning and on performing the mapping $x_t \to y_t$ that involves the peaking finding by (48). In contrast, a F-architecture implements the mapping $x_t \to y_t$ directly by a parametric model with a much reduced computing cost. However, there are two disadvantages in F-architecture. First, there is no consideration on noise. Second, as discussed at the end of Section II-C, there is no concept on principal or minor components such that the selection of the k best factors is meaningless.

In a BI-architecture, the advantages of a B-architecture and of a F-architecture are both retained via a parametric $q(x_t|y_t, y_{t-1})$ and a parametric $p(y_t|x_t, y_{t-1})$ in a tradeoff way. Moreover, a parametric $p(y_t|x_t, y_{t-1})$ also helps to regularize $q(x_t|y_t, y_{t-1})$ to avoid over-fitting.

We further investigate the BI-architecture part of (75) by starting at a special case that

$$p(y_t|x_t)=\delta(y_t-f^{\circ}(Wx_t+\mu)),\quad q(x_t|y_t)=G(x_t|Ay_t,\Sigma)$$

$$q(y_t|\theta_y)=\begin{cases}\displaystyle\prod_{j=1}^{k}[p_j\delta(y_t^{(j)}-1)+(1-p_j)\delta(y_t^{(j)})]\\ p_j=\dfrac{1}{1+e^{-c_j}}, & \text{(a) Bernoulli,}\\ G(y_t|\mu,\Lambda), & \text{(b) Gaussian,}\\ \text{By Item (C) in Table III,}\\ \qquad\text{(c) Independent non-Gaussian.}\end{cases}\tag{89}$$

In this case, the algorithm (75) is simplified as follows.

Step 1) $\hat{y}_t=f^{\circ}(W\overline{x}_t+\mu),\quad e_t=\overline{x}_t-A\hat{y}_t.$

Step 2)

$$\begin{aligned}z_q(t)&=z_q(t-1)+\delta z_q(t),\\ \gamma_t^b&=\frac{1}{t}-(1-i_R)\frac{\delta z_q(t)}{z_q(t)}\\ \delta z_q(t)&=G(e_t|0,\sigma^2)\\ &\quad\times\begin{cases}\displaystyle\prod_{j=1}^{k}p_j^{\hat{y}_t^{(j)}}(1-p_j)^{1-\hat{y}_t^{(j)}} & \text{(a) Bernoulli}\\ G(\hat{y}_t|m,\Lambda), & \text{(b) Gaussian}\\ q(\hat{y}_t|\theta_y), & \text{(c) Non-Gaussian.}\end{cases}\end{aligned}$$

Step 3)

a) For Bernoulli

$$\begin{aligned}c^{new}&=c^{old}+\zeta_y\gamma_t^b\varepsilon_t\\ \varepsilon_t^{(j)}&=y_t^{(j)}-p_j.\end{aligned}$$

b) For Gaussian

$$\begin{aligned}m^{new}&=m^{old}+\zeta_y\gamma_t^b\varepsilon_t\\ \varepsilon_t&=y_t-\mu^{old}\\ \Lambda^{new}&=(1-\zeta_y\gamma_t^b)\Lambda^{old}\\ &\quad+\zeta_y\gamma_t^b\,\text{diag}[\varepsilon_t\varepsilon_t^T+i_Rh_y^2I].\end{aligned}$$

c) For non-Gaussian

$$\begin{aligned}\theta_y^{new}&=\theta_y^{old}+\zeta_y\gamma_t^b\\ &\quad\cdot\left\{\delta\theta_y+0.5i_Rh_y^2\nabla_{\theta_y}\,\text{Tr}\left[\frac{\partial^2\ln q(y|\theta_y)}{\partial y\partial y^T}\right]\right\}\\ &\quad\text{with }\delta\theta_y\text{ given by Item (C) in Table III.}\end{aligned}$$

Step 4)

$$\begin{aligned}A^{new}&=A^{old}+\zeta_{x|y}\gamma_t^b[e_t\hat{y}_t^T-i_Rh_y^2A^{old}]\\ \Sigma^{new}&=(1-\zeta_{x|y}\gamma_t^b)\Sigma^{old}\\ &\quad+\zeta_{x|y}\gamma_t^b[e_te_t^T+i_R(h_x^2I+h_y^2AA^T)].\end{aligned}$$

Step 5)

$$\begin{aligned}\psi(y)&=A^{old\,T}\Sigma^{old\,-1}e_t,\\ D_f(y)&=\nabla_yf^{\circ}(y)\\ \phi(y)&=\begin{cases}\left[\dfrac{p_1^{old}}{1-p_1^{old}},\cdots,\dfrac{p_k^{old}}{1-p_k^{old}}\right]^T & \text{(a) Bernoulli}\\ -\Lambda^{old\,-1}(y-\mu^{old}), & \text{(b) Gaussian}\\ \nabla_y\ln q(y|\theta_y), & \text{(c) nonGaussian}\end{cases}\\ y_t^e&=D_f\hat{y}_t)[\phi(\hat{y}_t)+\psi(\hat{y}_t)]\\ W^{new}&=W^{old}+\zeta_{y|x}\gamma_t^by_t^e\overline{x}_t^T\\ \mu^{new}&=\mu^{old}+\zeta_{y|x}\gamma_t^by_t^e.\end{aligned}\tag{90}$$

Similar to the discussions made after (75), the above (90) is implemented via either normalization learning when $i_R=0$ or data smoothing learning when $i_R=1$. Again, both cases reduce to empirical learning when either $i_R=0$, $\gamma_t^b=1$ or $i_R=1$, $h_x=0$, $h_y=0$.

We can further understand the above algorithm by starting at the simplest special case that $q(y_t)$ is Bernoulli with $q_j=0.5$ and $z_q(t)$ is approximately regarded to be irrelevant to A, Σ. In this case, (65) becomes equivalent to

$$\begin{aligned}H_t(\theta,\mathbf{k})&=-0.5\ln|\Sigma|-k\ln 2,\\ \Sigma&=\frac{1}{t}\sum_{\tau=1}^{t}[\overline{x}_t-Af^{\circ}(W\overline{x}_t+\mu)]\\ &\quad\cdot[\overline{x}_t-Af^{\circ}(W\overline{x}_t+\mu)]^T,\quad\text{or}\\ \text{When }\Sigma=\sigma^2I,&\quad H_t(\theta,\mathbf{k})=-0.5d_x\ln\sigma^2-k\ln 2,\\ \sigma^2&=\frac{1}{t}\sum_{\tau=1}^{t}||\overline{x}_t-Af^{\circ}(W\overline{x}_t+\mu)||^2.\end{aligned}\tag{91}$$

Thus, $\max H_t(\theta,\mathbf{k})$ is equivalent to $\min\sigma^2$. Further letting $A=W^T$, it becomes equivalent to the least mean square error reconstruction (LMSER) learning that was firstly proposed with both batch and adaptive gradient algorithms provided in [75] and [82]. Moreover, it was also first discovered in [75] and [82] that with a sigmoid nonlinearity $f(r)$ it can automatically break the symmetry of the components in the subspace. Three years later, the LMSER learning and its adaptive algorithm given in [75] and [82] have been directly adopted to implement ICA with promising results by the authors of [37] under the name of nonlinear PCA.

Two direct extensions of the LMSER learning are obtained by relaxing $A=W^T$ and $\Sigma=\sigma^2I$. Also, it follows from (91) that we have the following simple criterion for selecting a best number k^* as the dimension of y_t:

$$\begin{aligned}\min_k J(k),\quad J(k)&=0.5d_x\ln\sigma^2+k\ln 2,\quad\text{or}\\ J(k)&=0.5\ln|\Sigma|+k\ln 2\end{aligned}\tag{92}$$

which is an open problem that has not be studied in [75].

Further extensions of the LMSER learning include 1) considering either the role of $z_q(t)$ by setting $i_R=0$ in (90) for normalization learning or the role of h_x by setting $i_R=1$ in (90) for data smoothing learning and 2) relaxing the constraint $q_j=0.5$ and letting $q(y|\theta_y)$ to be either of the three choices in (89) such that learning is made with automated model selection or structural minimization.

The general case of (90) can be understood from the term $(1/t)\sum_{\tau=1}^{t}\ln q(\overline{x}_\tau|\hat{y}_\tau,\hat{\xi}_\tau)+(1/t)\sum_{\tau=1}^{t}\ln q(\hat{y}_\tau|\hat{\xi}_\tau)$ which is the major part of both (65) and (70). From the fact that $(1/t)\sum_{\tau=1}^{t}\ln q(\hat{y}_\tau|\hat{\xi}_\tau)$ is also the major part of (66), we know that its role is to implement ICA or TICA. This point can also be observed from the existence of a term $\phi(y)$ in Step 5, which helps to reduce couplings among the components of y_t. Moreover, as discussed above, the term $(1/t)\sum_{\tau=1}^{t}\ln q(\overline{x}_\tau|\hat{y}_\tau,\hat{\xi}_\tau)$ implements LMSER-like learning that also performs ICA from the perspective of nonlinear PCA. Thus, the general case of (90) should perform ICA-like tasks by combining the two types of features.

Specifically, the forward $f^{\circ}(Wx_t+\mu)$ performs an ICA mapping such that the backward linear mapping $\overline{x}_t=Ay_t$ can best

reconstruct x_t. When the dimension d_x of x_t is larger than the dimension k of y_t, there will be $\binom{d_x}{k}$ choices to select the k components of y_t. The k components are regarded as principal or important in a sense that y_t can give a best reconstruction to x_t. In this case, we say that y_t retains the principal information of x_t. This role is similar to principal component analysis (PCA) that extracts a k-dimension subspace spanned by k principal components. But there is no such a role in the existing ICA approaches which only provide a k-dimension subspace spanned by any k independent factors. In other words, (90) implements a type of task that can be called principal ICA (P-ICA) or LMSER+ICA.

Similar to (92), we can also have

$$\min_k J(k) = 0.5\ln|\Sigma| + \begin{cases} -\frac{1}{t}\sum_{\tau=1}^{t}\sum_{j=1}^{k}[\hat{y}_\tau^{(j)}\ln p_j \\ \quad +(1-\hat{y}_\tau^{(j)})\ln(1-p_j)] & \text{Choice (a)} \\ 0.5(k\ln 2\pi + \ln|\Lambda| + k), & \text{Choice (b)} \\ -\frac{1}{t}\sum_{\tau=1}^{t}\sum_{j=1}^{k}\ln p(\hat{y}_\tau|\theta_y), & \text{Choice (c).} \end{cases} \tag{93}$$

The above LMSER-like learning, PICA and LMSER+ICA learning can all be further extended along two lines. One is to consider that $q(x|y,\xi), q(y|\xi)$ are both Bernoulli densities that are given and updated as in Item (B) of Table III, which leads to a case that is equivalent to the one layer deterministic Helmholtz machine learning [30], [21], [22], as previously discussed in [68], [63]. The other direction is made toward to temporal situations, which are covered by the BI-architecture part in (75).

C. Return Prediction, Macroeconomic Modeling, and Portfolio Management

In addition to directly implementing APT financial analyses as previously discussed, we can also make other financial applications in help of making APT analyses as a pre-stage. Three examples are given as follows.

- **Time Series Prediction:** After setting up an independent state-space model (13) with $\Lambda = I$, we can use x_{t-1} to get $\hat{y}_{t-1}$, e.g., by $\hat{y}_{t-1} = [A^T\Sigma^{-1}A+I]^{-1}(A^T\Sigma^{-1}x_{t-1} + B\hat{y}_{t-2})$ in the Gaussian TFA model. Then, we get $B\hat{y}_{t-1}$ which in turn provides a prediction $\overline{x}_t = AB\hat{y}_{t-1}$. This prediction method can be applied, e.g., to predict the return movement of securities.
- **Macroeconomic Prediction via Capital Market:** We can build a macroeconomic prediction model in two steps. First, we use statistical APT on the security returns x_t to get its hidden factors y_t. Second, we believe that the current economic situation is reflected in the capital market, and thus it is able to predict the macroeconomic indexes z_t via the current capital market. Therefore, we set up a regression $z_t = Cy_t + \epsilon_t$ based on the basic factors y_t of the capital market obtained via APT.
- **Portfolio Management:** Using an APT analysis as the first step, we can make portfolio management under the control of the discovered hidden factors behind the securities in the portfolio.

We consider a portfolio of risk securities with returns $x_t^{(j)}$, $j = 1,\cdots,m$ and a risk-free bond with return r^f, where the return is defined as $x_t^{(j)} = (p_t^{(j)} - p_{t-1}^{(j)})/p_{t-1}^{(j)}$. Moreover, we distribute α_t percentage of capital on risk securities and thus $1-\alpha_t$ percentage of capital on the risk-free bond. In this case, the return of the portfolio is given by

$$R_t = (1-\alpha_t)r^f + \alpha_t\sum_{j=1}^{m}\beta_t^{(j)}x_t^{(j)}, \quad \text{subject to} \begin{cases} \alpha_t > 0, 1 \geq \beta_t^{(j)} \geq 0, \\ \sum_{j=1}^{m}\beta_t^{(j)} = 1, \quad \text{Choice (a)} \\ \sum_{j=1}^{m}\beta_t^{(j)} = 1, \text{no constraint on} \\ \quad \alpha_t, \beta_t^{(j)}, \qquad \text{Choice (b).} \end{cases} \tag{94}$$

In the choice (a), short of a risk security is not permitted but borrowing from the risk-free bond is allowed (i.e., we can have $1-\alpha_t < 0$ or $\alpha_t \geq 1$). While in the choice (b), short of a risk security is permitted.

Our purpose is to maximize the return R_t by adaptively controlling the weights $\beta_t^{(j)}$ as time t goes. Any change on $\beta_t^{(j)}$ leads to a transaction that incurs a cost return

$$\begin{aligned} c_t &= -\alpha_t\sum_{j=1}^{m} r_c|\beta_t^{(j)} - \beta_{t-1}^{(j)}|p_t^{(j)}/p_{t-1}^{(j)} \\ &= -\alpha_t\sum_{j=1}^{m} r_c|\beta_t^{(j)} - \beta_{t-1}^{(j)}|(1+x_t^{(j)}) \end{aligned}$$

where r_c is the rate of transaction cost. Thus, we have the following adjusted return:

$$\begin{aligned} \tilde{R}_t = (1-\alpha_t)r^f + \alpha_t\sum_{j=1}^{m}[&\beta_t^{(j)}x_t^{(j)} \\ &- r_c|\beta_t^{(j)} - \beta_{t-1}^{(j)}|(1+x_t^{(j)})]. \end{aligned} \tag{95}$$

We want not only to maximize the adjusted return $\tilde{R}_t$ but also to minimize its uncertainty or called volatility, which is measured by its variance $V(\tilde{R}_t)$. This purpose is implemented by

$$\max_{\psi,\phi} S_p, \quad S_p = \frac{M(\tilde{R}_t)}{\sqrt{V(\tilde{R}_t)}}, \quad M(\tilde{R}_t) = \frac{1}{T}\sum_{t=0}^{T-1}\tilde{R}_t$$

$$V(\tilde{R}_t) = \frac{1}{T}\sum_{t=0}^{T-1}[\tilde{R}_t - M(\tilde{R}_t)]^2,$$

$$\text{Subject to} \begin{cases} \alpha_t = g(y_t,\psi) > 0 \\ \beta_t^{(j)} = e^{\hat{\xi}_t^{(j)}} \Big/ \sum_{r=1}^{m} e^{\hat{\xi}_t^{(r)}}, \\ \hat{\xi}_t = f(y_t,\phi), \quad \text{for Choice (a),} \\ \alpha_t = g(y_t,\psi), \quad \beta_t^{(j)} = \hat{\xi}_t^{(j)} \\ \hat{\xi}_t = f(y_t,\phi), \quad \text{for Choice (b).} \end{cases} \tag{96}$$

The above S_p is a modification of the well-known Sharpe ratio $S_p = M(R_t)/\sqrt{V(R_t)}$ that has been widely used in the finance literature for evaluating the performance of a portfolio [53], [52]. In recent years, several efforts have been made on

maximizing the Sharpe ratio by adjusting the weights $\beta_t^{(j)}$, instead of just being used for a performance evaluation [42], [15]. In those existing efforts, the weights $\beta_t^{(j)}$ either are constants or depend directly on the security returns x_t.

In contrast, $\beta_t^{(j)}$ in (96) are functions of the independent factor vector y_t in (13) and we learn the best weights $\beta_t^{(j)}$ after we first implement the temporal APT model (13) and then make the mapping $x_t \to y_t$. Specifically, both α_t, $\beta_t^{(j)}$ in (96) are modeled with both $f(y_t, \phi)$ and $g(y_t, \psi)$ implemented by any forward network, e.g., a mixture of expert or its special case—the extended normalized RBF net [66]. The simplest case is to use linear functions

$$g(y_t, \psi) = \psi^T y_t + \psi_0, \quad f(y_t, \phi) = \phi^T y_t + \phi_0. \tag{97}$$

We can indirectly adjust ϕ, ψ to get optimal β_t, α_t that change with t under the control of y_t. In implementation, the maximization can be made simply by using gradient ascent method.

The above method will deteriorate when the variance $V(R_t)$ varies with t. One solution is to use the ARCH [26] or GARCH [11] to estimate the time-varying variance $V(R_t)$ and then make learning adaptively

$$\begin{aligned} \phi^{new} &= \phi^{old} + \zeta \frac{\partial(\tilde{R}_t/\sqrt{V(R_t)})}{\partial \phi}, \\ \psi^{new} &= \psi^{old} + \zeta \frac{\partial(\tilde{R}_t/\sqrt{V(R_t)})}{\partial \psi}, \\ \zeta &> 0 \text{ is a stepsize.} \end{aligned} \tag{98}$$

D. Macroeconomics Modulated Independent State-Space Model

For a market of m risk securities which form a vector $\mathbf{x_t}$ in R^m, if we can get a portfolio via the m weights of $\beta^{(j)}$ which also form a vector $\boldsymbol{\beta}$ in R^m such that

$$\begin{aligned} &\sum_{j=1}^{m} \beta^{(j)} x_t^{(j)} = \mathbf{x_t}^T \boldsymbol{\beta} > 0, \text{ with} \\ &\sum_{j=1}^{m} \beta^{(j)} = \mathbf{1}^T \boldsymbol{\beta} = 0, \qquad \mathbf{1}^T = [1, 1, \cdots, 1] \end{aligned} \tag{99}$$

it is said that we have an arbitrage chance of getting a profit with a zero investment. All the possible zero investments form a $m-1$ dimensional hyperplane that passes the origin of R^m and is orthogonal to the vector $\mathbf{1}$. Thus, no arbitrage chance in an equilibrium market means that $\mathbf{x_t}$ in equilibrium is orthogonal to the same $m-1$ dimensional hyperplane, i.e., $\mathbf{x_t}^T \boldsymbol{\beta} = 0$. As a result, we get $\mathbf{x_t} = c\mathbf{1}$ with c being an arbitrary constant. That is, each security gets a same return, and thus it is not a much interesting case. So, strictly speaking, there should be some arbitrage chance theoretically, though such a chance is difficult to discover or use due to fast or random movement of $\mathbf{x_t}$.

The APT theory imposes an arbitrage constraint on the factor model $x_t^{(j)} = \sum_{i,j} a_{i,j} y_t^{(i)} + e_t^{(j)}$ in (8) or equivalently $x_t = Ay_t + e_t$ in (1). That is, we have

$$\begin{aligned} &\sum_{j=1}^{m} \beta^{(j)} a_{i,j} = \mathbf{a}_i^T \boldsymbol{\beta} = 0, \quad i = 1, \cdots, k, \text{ with} \\ &\mathbf{1}^T \boldsymbol{\beta} = 0 \end{aligned} \tag{100}$$

where $\mathbf{a}_i^T = [a_{i,1}, \cdots, a_{i,m}]$. Similarly, requiring (100) to hold for any zero investment leads to $\mathbf{a}_i^T = c_i \mathbf{1}$ for constants c_i, $i = 1, \cdots, m$. So, strictly speaking, an equilibrium market via $x_t = Ay_t + e_t$ with no arbitrage chance does not exist in the sense of considering all the possible zero investments, though it may exist when we only consider one or a subset of specific zero investments.

Therefore, we relax the condition of an equilibrium market from no arbitrage chance to that there is an equilibrium factor model $x_t = Ay_t + e_t$ such that the series of e_t consists of i.i.d samples from a density that is independent from y_t, and the elements of y_t are independent to each other but may be time-varying as t.

Moreover, we use model $y_t = By_{t-1} + \varepsilon_t$ and thus get the state-space model (13), which are then implemented in various cases as given in Section III-A. Particularly, for Gaussains $G(\varepsilon_t|0, \Lambda)$ and $G(e_t|0, \Sigma)$, we use (78) for the modeling via implementing Gaussian TFA.

Furthermore, we can also take macroeconomic indexes, which are usually observable, into the modeling of a capital market as follows:

$$\begin{aligned} &y_t = By_{t-1} + Hz_{t-1} + \varepsilon_t, \quad x_t = Ay_t + e_t \\ &z_t = Cy_t + Ev_t + \epsilon_t \\ &\varepsilon_t, e_t, \epsilon_t \text{ are white noises and independent} \\ &\text{from each other} \\ &\varepsilon_t \text{ is independent from both } z_{t-1} \text{ and } y_{t-1} \\ &e_t, \epsilon_t \text{ are independent from } y_t, v_t \end{aligned} \tag{101}$$

where z_t consists of a number of macroeconomic indexes, and v_t consists of a number of known nonmarket-factors that affect the macro-economy. Specifically, Hz_{t-1} describes the indirect effect of the macroeconomic indexes to the security market via the hidden factors y_t, and Cy_t describes the feedback effect of the market to the macroeconomic indexes. Thus, we call (101) *macroeconomics modulated independent state-space model*. We belive that the model (101) describes a capital market via both a short-term dynamics and a long-term dynamics. In the short-term dynamics, x_t, y_t and perhaps z_t move to reach an equilibrium in the sense that the series of e_t, ε_t, ϵ_t become stationary white noises, while the parameters B, H, A, C, E and statistics of e_t, ε_t, ϵ_t can be relatively regarded as constants due their slow changing. In a long-term dynamics, the parameters B, H, A, C, E and statistics of e_t, ε_t, ϵ_t are all in changing to cohere the current equilibrium.

The process of the long-term dynamics is the process of learning. With H fixed, Hz_t acts as a constant and can be regarded as a part of the mean of ε_t. Thus, we can make learning on the first two equations in (101) in the same way as in (78). Moreover, $z_t = Cy_t + Ev_t + \epsilon_t$ can be handled in the way similar to $x_t = Ay_t + e_t$. Furthermore, the task of estimating H is a linear regression problem when y_t, y_{t-1}, B are fixed. Particularly, for $G(\varepsilon_t|Hz_{t-1}, \Lambda)$, $G(e_t|0, \Sigma_x)$ and $G(\epsilon_t|0, \Sigma_z)$ we can modify (78) for implementing (101) as follows:

Step 1)

$$\begin{aligned} \hat{y}_t = &[\Lambda^{-1} + A^T \Sigma_x^{-1} A + C^T \Sigma_z^{-1} C]^{-1} \\ &\cdot [A^T \Sigma_x^{-1} \overline{x}_t + C^T \Sigma_z^{-1} (\overline{z}_t - Ev_t) \\ &+ \Lambda^{-1} (By_{t-1} + H\overline{z}_{t-1})] \end{aligned}$$

$$\varepsilon_t = \hat{y}_t - B\hat{y}_{t-1} - H\overline{z}_{t-1}$$
$$e_t = \overline{x}_t - A\hat{y}_t, \quad \epsilon_t = \overline{z}_t - C\hat{y}_t - Ev_t;$$

Step 2)

$$\delta z_q(t) = G(\varepsilon_t|0,\Lambda^{old})G(e_t|0,\Sigma_x^{old}) \cdot G(\epsilon_t|0,\Sigma_z^{old})$$
$$\gamma_t^b = \frac{1}{t} - (1-i_R)\frac{\delta z_q(t)}{z_q(t)}$$
$$z_q(t) = z_q(t-1) + \delta z_q(t);$$

Step 3)

$$B^{new} = B^{old} + \eta\gamma_t^b \text{ diag}[\varepsilon_t\hat{y}_{t-1}^T],$$
$$H^{new} = H^{old} + \eta\gamma_t^b \text{ diag}[\varepsilon_t\overline{z}_{t-1}^T]$$
$$\Lambda^{new} = (1-\eta\gamma_t^b)\Lambda^{old} + \eta\gamma_t^b \cdot \text{diag}[\varepsilon_t\varepsilon_t^T + i_R h_y^2 I]$$
$$E^{new} = E^{old} + \eta\gamma_t^b \text{ diag}[\epsilon_t v_t^T];$$

Step 4)

$$A^{new} = A^{old} + \eta\gamma_t^b[e_t\hat{y}_t^T - i_R h_y^2 A^{old}]$$
$$\Sigma_x^{new} = (1-\eta\gamma_t^b)\Sigma_x^{old} + \eta\gamma_t^b[e_te_t^T + i_R(h_x^2 I + h_y^2 A^{\text{old}}A^{\text{old}T})];$$

Step 5)

$$C^{new} = C^{old} + \eta\gamma_t^b[\epsilon_t\hat{y}_t^T - i_R h_y^2 C^{old}]$$
$$\Sigma_z^{new} = (1-\eta\gamma_t^b)\Sigma_z^{old} + \eta\gamma_t^b[\epsilon_t\epsilon_t^T + i_R(h_z^2 I + h_y^2 C^{\text{old}}C^{\text{old}T})] \quad (102)$$

where $\{\overline{z}_t\}_{t=1}^N$ are a set of observations on z_t that consists of a number of the macroeconomic indexes.

1) Remarks:

a) Similar to the discussions made after (90), (102) is implemented via either normalization learning when $i_R = 0$ or data smoothing learning when $i_R = 1$. Also, both cases degenerate to empirical learning when either $i_R = 0$, $\gamma_t^b = 1$ or $i_R = 1$, $h_x = 0$.

b) When making data smoothing learning, we can update h_x, h_z, h_y via either simulated annealing or (81) which now becomes

$$h_x^{\text{new2}} \approx \frac{2h_{x,0}^2}{1+\sqrt{1+4h_{x,0}^2 d_x^{-1}\text{Tr}[\sum_x^{-1}]}}$$
$$h_z^{\text{new2}} \approx \frac{2h_{z,0}^2}{1+\sqrt{1+4h_{z,0}^2 d_z^{-1}\text{Tr}[\sum_z^{-1}]}}$$
$$h_y^{\text{new2}} \approx \frac{2h_{y,0}^2}{1+\sqrt{1+4h_{y,0}^2 k - 1\text{Tr}[A^T\sum_z^{-1}A + C^T\sum_x^{-1}C + \Lambda^{-1}]}}$$
$$h_{x,0}^2 = \frac{1}{d_x}\sum_{\tau=1}^N\sum_{t=1}^N \gamma_{t,\tau}\|x_t - x_\tau\|^2$$
$$h_{y,0}^2 = \frac{1}{k}\sum_{\tau=1}^N\sum_{t=1}^N \gamma_{t,\tau}\|y_t - y_\tau\|^2$$
$$h_{z,0}^2 = \frac{1}{d_z}\sum_{\tau=1}^N\sum_{t=1}^N \gamma_{t,\tau}\|z_t - z_\tau\|^2$$

$$\gamma_{t,\tau} = \frac{e^{-0.5[\frac{\|x_t-x_\tau\|^2}{h_x^{\text{old2}}} + \frac{\|y_t-y_\tau\|^2}{h_y^{\text{old2}}} + \frac{\|z_t-z_\tau\|^2}{h_z^{\text{old2}}}]}}{\sum_{\tau=1}^N\sum_{t=1}^N \gamma_{t,\tau}e^{-0.5[\frac{\|x_t-x_\tau\|^2}{h_x^{\text{old2}}} + \frac{\|y_t-y_\tau\|^2}{h_y^{\text{old2}}} + \frac{\|z_t-z_\tau\|^2}{h_z^{\text{old2}}}]}} \quad (103)$$

where d_z is the number of macroeconomic indexes.

c) By simply setting $E = 0$ in (101) and (102), we can ignore v_t when it is not available. Moreover, by setting $C = 0$ we can ignore the effect of the market to the effect of macroeconomic indexes, while by setting $H = 0$ we can ignore the macroeconomic indexes to the market.

d) We can also take in consideration the fact that securities are divided into n groups, with each group affected by specific factors and macroeconomic indexes. Usually, the interactions between groups are much weaker and thus can be ignored. In such cases, we can impose the constraint that A, B, C, H are block matrices of the form $U = \text{diag}[U_1, \cdots, U_n]$.

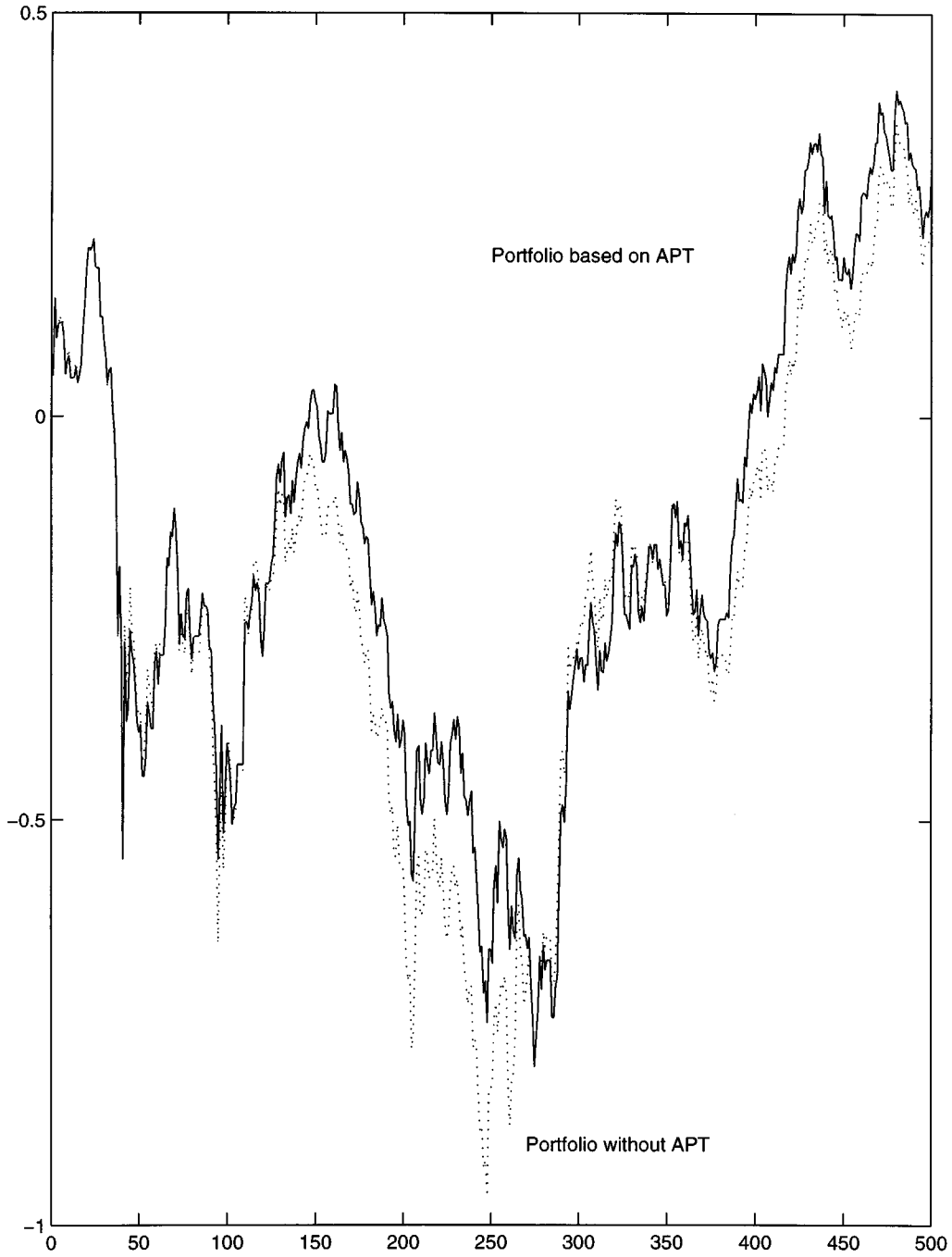
E. Experimental Illustrations

Experiments on temporal model (13) can be found in [63], [14], [61], either with real factors by the Gaussian TFA algorithm (79) and temporal ICA, or with binary factors by the Bernoulli FA and independent HMM algorithm (84). Here, we give two more illustrations on financial analyses.

- **Illustration on the APT-Based Portfolio Management:** We consider to set up a portfolio by (96) such that it is managed dynamically by $\beta_t^{(j)}$. The portfolio consists of four compound securities of Hong Kong Heng Seng indexes on finance, utilities, properties and commerce and industry, denoted by $x_t^{(j)}$, $j = 1,2,3,4$. The data consists of the closing value of each day from 1 Jan 1990 to 27 July 1999. The first 2000 samples are used for training the value of β_t by (96). The remaining 500 samples are used for testing the results. The transaction costs are ignored.

In implementation, we first use the Gaussian TFA algorithm (79) on the temporal APT model (13) to get the hidden factor y_t from securities $x_t^{(j)}$, $j = 1,2,3,4$. Then, we use the obtained y_t in (96) to get $\hat{\xi}_t = \phi_y^T y_t + \phi_{y,0}$. Equation (96) is implemented in the batch way by a gradient ascending algorithm. To compare the advantage of using the hidden independent factors for controlling, we also run (96) on the same experiment with $\hat{\xi}_t = \phi_x^T x_t + \phi_{x,0}$, with x_t consisting of the original indexs. For simplicity, we denote (96) by Portfolio based on TFA for the case $\hat{\xi}_t = \phi_y^T y_t + \phi_{y,0}$, and by Portfolio without TFA for the case $\hat{\xi}_t = \phi_x^T x_t + \phi_{x,0}$.

Shown in Fig. 2(a) are the resulted relative returns by the Portfolio based on TFA and the Portfolio without TFA. It can be observed that the Portfolio based on TFA outperforms the Portfolio without TFA. Moreover, we can recursively get the portfolio price $p_t = p_{t-1}(1+R_t)$ with initial $p_0 = 1$. We plot each price curve of p_t in comparison with the normalized original indices such that all of them start at the same level that is one. As shown in Fig. 2(b), the Portfolio based on TFA outperform the Portfolio without TFA considerably, and also both outperform each of the original indexes. However, it should be noticed that this is only a preliminary result. The ignorance of transaction



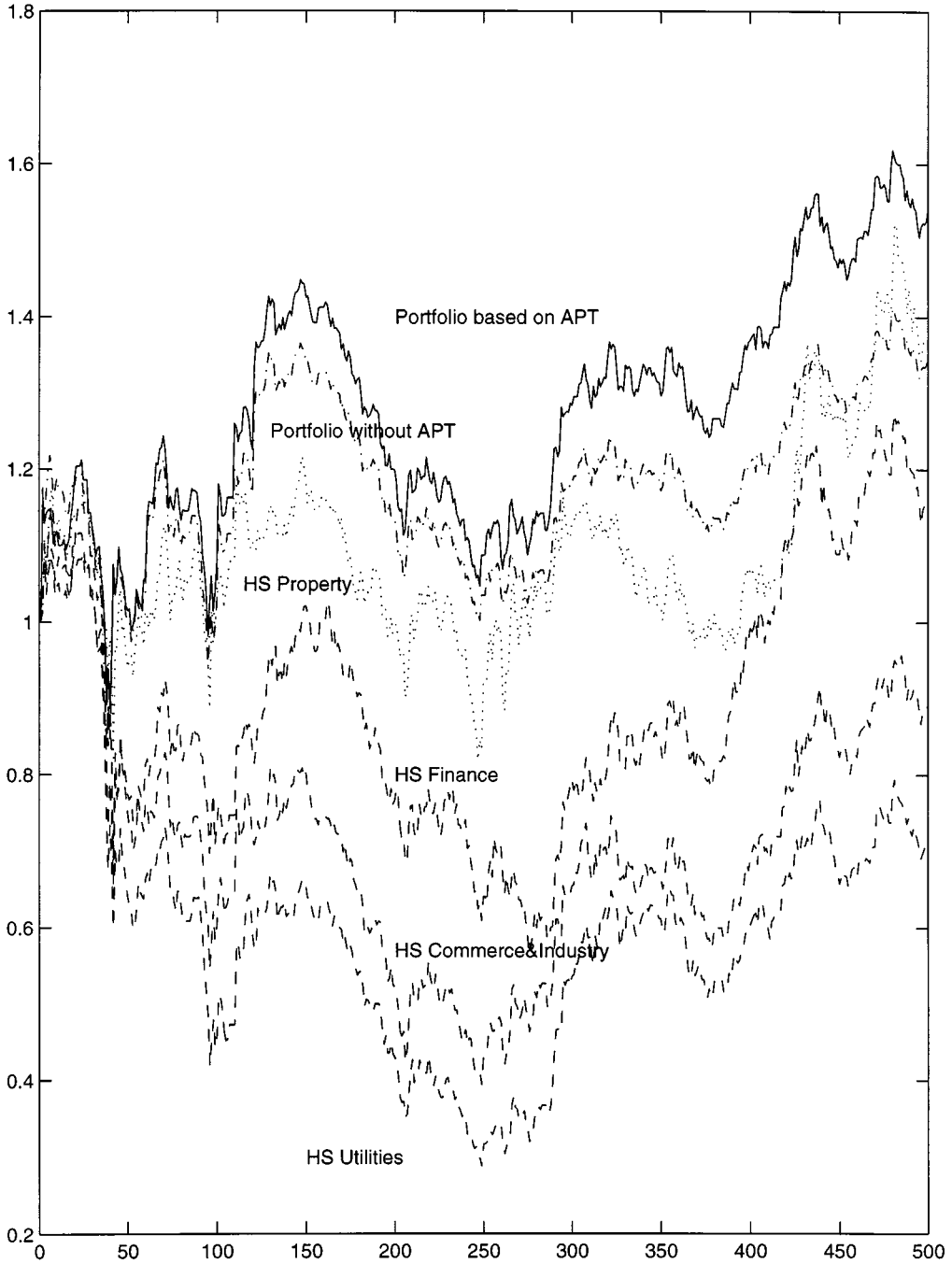
(a)

Fig. 2. Comparisons on Portfolio based on TFA and Portfolio without TFA. (a) The relative returns.

costs can be a severe limitation. Further experiments should be made with these costs taken in consideration.

- **Illustration on Modeling and Prediction of Macroeconomic Indexes via Securities:** Preliminary experiments are also demonstrated on the proposed two-step modeling in Section III-C for modeling macroeconomic indexes via securities in a capital market. Focusing on the issue of the higher order independence versus the second-order independence only, we omit the temporal relation and ignore the observation noise e_t, and implement the mapping $y_t = Wx_t$ by using the learned parametric mixture based ICA [69], [72], [79], i.e., the special case ii) of (88), in comparison with using PCA.

The experiments were made on the 340 real stocks of the S&P 500 since January 1973. In the same period, we consider ten macroeconomic indexes. The linear mapping $y_t = Wx_t$ maps the 340–dimensional x_t into the ten-dimensional factor vector y_t. Then, we use the least square approach to build up a linear repression $z_t^{(j)} = E_j y_t$ between each $z_t^{(j)}$ of the ten macroeconomic indexes and the ten basic factors y_t. Specifically, we use



(b)

Fig. 2. (*Continued*.) Comparisons on Portfolio based on TFA and Portfolio without TFA. (b) The portfolio price p_{t+1} with initial $p_0 = 1$ and the normalized indexes $\overline{x}_t^{(j)} = x_t^{(j)}/x_0^{(j)}$, j = 1, 2, 3, 4.

80% of data as a training set for getting A, E_j, and then use 20% of data as a testing set to get $\hat{y}_t$ by $y_t = Wx_t$ and to predict each $z_t^{(j)}$ of the ten macroeconomic indexes by $z_t^{(j)} = E_j\hat{y}_t$. The prediction performance is demonstrated via the mean square error $N^{-1}\sum_t \sum_{j=1}^{10}(z_t^{(j)} - E_j y_t)^2$ as shown in Table IV. The prediction results based on ICA outperform considerably the results based on PCA, which have demonstrated the feasibility of using a F-architecture-based ICA for a simplified APT implementation. However, this is only a preliminary result and further experiments should be made with error bars provided.

IV. Conclusion

The relationship between factor analysis and the well-known arbitrage pricing theory (APT) for financial market has been discussed, with a number of to-be-improved problems listed. The BYY ISS system and harmony learning principle, with a general adaptive learning procedure, have been suggested as a unified guide to tackle the problems systematically. New adaptive algorithms, regularization methods and model selection criteria are

TABLE IV
THE MEAN SQUARE ERROR (MSE) OF PREDICTIONS ON MACROECONOMIC INDEXES

Macroeconomic Indexes	PCA	ICA
Dow Jones Industrial Index	59007	57090
Dow Jones Average Index	5108.3	4787.6
S&P500 Average Index	580.2854	571.7302
US Customer Price Index	0.0785	0.075
US Industrial Production Index	0.4352	0.3764
US Civilian Employment	67395	66079
US Consumer Confidence Index	21.7832	21.5333
US Producer Price Index	0.155	0.151
US Total Business Sales	27729000	25057000
US Unemployment Rate	0.234	0.214

provided on various specific cases of each of three typical architectures, with applications to APT analyses for solving the listed to-be-improved problems. Moreover, other APT-based applications, namely macroeconomic prediction, optimal asset allocation, and macroeconomics modulated independent state-space model are also proposed.

ACKNOWLEDGMENT

The author thanks L. L. Tan and F. Yip for experiments in Section III-E.

REFERENCES

[1] S. Abeyskera and A. Mahajan, "A test of the APT in pricing UK stocks," *J. Accounting Finance*, vol. 17, pp. 377–391, 1987.
[2] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. 19, pp. 714–723, 1974.
[3] S.-I. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind separation of sources," in *Advances in Neural Information Processing*, D. S. Touretzky *et al.*, Eds. Cambridge, MA: MIT Press, 1996, vol. 8, pp. 757–763.
[4] T. W. Anderson and H. Rubin, "Statistical inference in factor analysis," in *Proc. 3rd Berkeley Symp. Math. Statist. Prob.*, vol. 5, Berkeley, 1956, pp. 111–150.
[5] A. Back and A. S. Weigend, "A first application of independent component analysis to extracting structures from stock returns," *Int. J. Neural Syst.*, vol. 8, pp. 473–484, 1997.
[6] A. Barron and T. Cover, "Minimum complexity density estimation," *IEEE Trans. Inform. Theory*, vol. 37, no. 4, pp. 1034–1054, 1991.
[7] H. B. Barlow, "Unsupervised learning," *Neural Comput.*, vol. 1, pp. 295–311, 1989.
[8] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, pp. 1129–1159, 1995.
[9] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, pp. 108–116, 1995.
[10] P. J. Bolland and J. T. Connor, "A constrained neural network Kalman filter for price estimation in high-frequency financial data," *Int.. J. Neural Syst.*, vol. 8, pp. 399–415, 1997.
[11] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econometrics*, vol. 31, pp. 307–327, 1986.
[12] H. Bozdogan, "Model selection and Akaike's information criterion: The general theory and its analytical extension," *PSYCHOMETRIKA*, vol. 52, pp. 345–370, 1987.
[13] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York: Wiley, 1997.
[14] Y. M. Cheung and L. Xu, "Further studies on temporal factor analysis," in *Proc. ICONIP'2000*, vol. 6, Taejon, Korea, Nov. 14–18, 2000, pp. 1371–1376.
[15] M. Choey and A. S. Weigend, "Nonlinear trading models through Sharpe ratio optimization," *Int. J. Neural Syst.*, vol. 8, pp. 417–431, 1997.
[16] A. Cichocki, S. C. Douglas, and S. Amari, "Robust techniques for independent component analysis (ICA) with noisy data," *Neurocomput.*, vol. 22, pp. 113–129, 1998.
[17] P. Comon, "Independent component analysis—A new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.
[18] G. Connor, "The three types of factor models: A comparison of their explanatory power," *Financial Analysts J.*, pp. 42–46, May/June 1995.
[19] G. Connor and R. Korajczyk, "Performance measurement with the arbitrage pricing theory: A new framework for analysis," *Journal of Financial Economics*, vol. 15, pp. 373–394, 1986.
[20] G. Connor, "A test for the number of factors in an approximate factor structure," *J. Finance*, vol. 48, pp. 1263–1291, 1993.
[21] P. Dayan and R. S. Zemel, "Competition and multiple cause models," *Neural Comput.*, vol. 7, pp. 565–579, 1995.
[22] P. Dayan and G. E. Hinton, "Varieties of Helmholtz machine," *Neural Networks*, vol. 9, pp. 1385–1403, 1996.
[23] D. Desieno, "Adding a conscience to competitive learning," in *Proc. IEEE Int. Conf. Neural Networks*, vol. I, 1988, pp. 117–124.
[24] L. Devroye *et al.*, *A Probability Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.
[25] P. Dhyymes, I. Friend, and B. Gultekin, "A critical reexamination of its empirical evidence on the arbitrage pricing theory," *J. Finance*, pp. 323–346, June 1984.
[26] R. F. Engle, "Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, pp. 987–1007, 1982.
[27] C. Fyfe, *Special issue on Independence and Artificial Neural Networks, Neurocomputing*, vol. 22, no. 1–3, 1998.
[28] M. Gaeta and J.-L. Lacounme, "Source separation without a priori knowledge: The maximum likelihood solution," in *Proc. EUSIPCO90*, 1990, pp. 621–624.
[29] F. Girosi *et al.*, "Regularization theory and neural architectures," *Neural Comput.*, vol. 7, pp. 219–269, 1995.
[30] G. E. Hinton *et al.*, "The wake-sleep algorithm for unsupervised learning neural networks," *Science*, vol. 268, pp. 1158–1160, 1995.
[31] H. Hotelling, "Simplified calculation of principal components," *Psychometrika*, vol. 1, pp. 27–35, 1936.
[32] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, pp. 181–214, 1994.
[33] M. I. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts," *Neural Networks*, vol. 8, pp. 1409–1431, 1995.
[34] P. de Jong, "The likelihood for a state space model," *Biometrica*, vol. 75, pp. 165–169, 1989.
[35] C. Jutten and J. Herault, "Independent component analysis versus principal component analysis," in *Proc. Europ. Signal Processing Conf. EUSIPCO88*, 1988, pp. 643–646.
[36] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, pp. 35–45, Mar. 1960.
[37] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks*, vol. 7, pp. 113–127, 1994.
[38] S. Kirkpatrick *et al.*, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[39] D. J. C. Mackey, "A practical Bayesian framework for backpropagation," *Neural Computation*, vol. 4, pp. 415–447, 1992.
[40] R. McDonald, *Factor Analysis and Related Techniques*. Hillsdale, NJ: Lawrence Erlbaum, 1985.
[41] J. Moody and L. Wu, "What is the 'true price'?—State space models for high frequency financial data," in *Progress in Neural Information Processing (ICONIP96)*. New York, 1996, pp. 697–704.
[42] ——, "Optimization of trading systems and portfolios," in *Proc. CIFEr97*, New York, 1997, pp. 300–307.
[43] E. Oja, *Subspace Methods of Pattern Recognition*. London, UK: Research Studies Press, 1983.
[44] E. Oja and A. Hyvarinen, "Blind signal separation by neural networks," in *Proc. ICONIP96*. New York, 1996, pp. 7–14.
[45] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[46] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood, and the EM algorithm," *SIAM Rev. 26*, pp. 195–239, 1984.
[47] S. Ross, "The arbitrage theory of capital assert pricing," *J. Economic Theory*, vol. 13, pp. 341–360, 1976.

[48] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.
[49] S. Ross, "Risk, return, and arbitrage," in *Risk and Return in Finance I*, I. Friend and J. Bicksler, Eds. Cambridge, MA, 1977.
[50] D. Rubi and D. Thayer, "EM algorithm for ML factor analysis," *Psychometrika*, vol. 57, pp. 69–76, 1976.
[51] E. Saund, "A multiple cause mixture model for unsupervised learning," *Neural Comput.*, vol. 7, pp. 51–71, 1995.
[52] W. F. Sharpe, C. J. Alexander, and J. V. Bailey, *Investment*. Englewood Cliffs, NJ: Prentice Hall, 1995.
[53] ——, "The Sharpe ration—Properly used, it can improve investment," *J. Portfolio Management*, pp. 49–58, 1994.
[54] C. Spearman, "General intelligence objectively determined and measured," *Amer. J. Psychol.*, vol. 15, pp. 201–293, 1904.
[55] M. Stone, "Cross-validation: A review," *Math. Operat. Statist.*, vol. 9, pp. 127–140, 1978.
[56] J. Timmer and A. S. Weigend, "Modeling volatility using state space models," *Int. J. Neural Syst.*, vol. 8, pp. 385–398, 1997.
[57] A. Taleb and C. Jutten, "Non-linearity source separation: The post-nonlinear mixtures," in *Proc. ESANN97*, Apr. 16–18, 1997, pp. 279–284.
[58] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC: V. H. Winston and Sons, 1977.
[59] L. Tong, Y. Inouye, and R. Liu, "Waveform-preserving blind estimation of multiple independent sources," *IEEE Trans. Signal Processing*, vol. 41, pp. 2461–2470, 1993.
[60] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[61] L. Xu, "Binary factor analysis, hidden Markov model, and unsupervised mining of knowledge production rules," in *Proc. Inl. Joint Conf. Neural Networks*, Washington, DC, July 15–19, 2001.
[62] ——, "Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models," *Int. J. Neural Syst., special issue*, vol. 11, no. 1, pp. 43–49, 2001.
[63] ——, "Temporal BYY learning for state space approach, hidden markov model and blind source separation," *IEEE Trans. Signal Processing*, vol. 48, pp. 2132–2144, 2000.
[64] ——, "BYY system and theory for statistical learning: Best harmony, data smoothing, and model selection," *Neural, Parallel, and Scientific Computations*, vol. 8, pp. 55–82, 2000.
[65] ——, "Best harmony learning," in *Lecture Notes in Computer Science 1983*. New York: Springer-Verlag, 2000, pp. 116–125.
[66] ——, "RBF nets, mixture experts, and bayesian ying-yang learning," *Neurocomput.*, vol. 19, no. 1–3, pp. 223–257, 1998.
[67] ——, "Bayesian ying-yang learning theory for data dimension reduction and determination," *J. Comput. Intell. Finance*, vol. 6, no. 5, pp. 6–18, 1998.
[68] ——, "Bayesian kullback ying-yang dependence reduction theory," *Neurocomput.*, vol. 22, no. 1–3, pp. 81–112, 1998.
[69] L. Xu, C. C. Cheung, and S.-I. Amari, "Learned parametric mixture based ICA algorithm," *Neurocomputing*, vol. 22, no. 1–3, pp. 69–80, 1998.
[70] L. Xu, "Bayesian ying-yang learning-based ICA models," in *Proc. 1997 IEEE Signal Processing Society Workshop*, FL, Sept. 24–26, 1997, pp. 476–485.
[71] ——, "Bayesian ying-yang machine, clustering and number of clusters," *Pattern Recognition Lett.*, vol. 18, no. 11–13, pp. 1167–1178, 1997.
[72] L. Xu, H. H. Yang, and S.-I. Amari, "Signal source separation by mixtures accumulative distribution functions or mixture of bell-shape density distribution functions," , Japan, Apr. 10, 1996, Research Proposal, presented at *FRONTIER FORUM* (speakers: D. Sherrington, S. Tanaka, L. Xu & J. F. Cardoso), organized by S. Amari, S. Tanaka, and A. Cichocki.
[73] L. Xu, "A unified learning scheme: Bayesian-Kullback ying-yang machine," *Advances in Neural Information Processing Systems*, vol. 8, pp. 444–450, 1996.
[74] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," in *Advances in Neural Information Processing Systems*, J. D. Cowan *et al.*, Eds: MIT Press, 1995, vol. 7, pp. 633–640.
[75] L. Xu, "Least mean square error reconstruction for self-organizing neural-nets," *Neural Networks*, vol. 6, pp. 627–648, 1993.
[76] L. Xu, "Temporal BYY learning and its applications to extended Kalman fildering, hidden Markov model, and sensor-motor integration," in *Proc. IJCNN99*, vol. 2, pp. 949–954.
[77] ——, "Bayesian ying-yang system and a theory as a unified statistical learning approach: Temporal modeling for temporal perception and control," in *Proc. ICONIP98*, vol. 2, Kitakyushu, Japan, pp. 877–884.
[78] ——, "BYY system and theory for statistical learning: Best harmony, data smoothing, and model selection," in *Proc. 1999 Chinese Conf. NNSP*, 1999, pp. 12–29.
[79] L. Xu, C. C. Cheung, J. Ruan, and S.-I. Amari, "Nonlinearity and separation capability: Further justification for the ICA algorithm with a learned mixture of parametric densitites," in *Proc. ESANN97*, Bruges, Belgium, Apr. 16–18, 1997, pp. 291–296.
[80] L. Xu, "Bayesian-Kullback coupled ying-yang machines: Unified learning and new results on vector quantization," *Proc. ICONIP95*, pp. 977–988, Oct. 30–Nov. 3, 1995.
[81] L. Xu, M. I. Jordan, and G. E. Hinton, "A modified gating network for the mixtures of experts architectures," in *Proc. WCNN'94* San Diego, CA, 1994, vol. 2, pp. 405–410.
[82] L. Xu, "Least mean square error reconstruction for self-organization: (I) Multilayer neural-nets and (II) Further theoretical and experimental studies on one-layer nets," in *Proc. IJCNN91*, Singapore, 1991, pp. 2363–2373.

Lei Xu (SM'94–F'01) received the Ph.D degree from Tsinghua University, China, in 1987.

He is a Professor of Department Computer Science and Engineering at the Chinese University of Hong Kong (CUHK). He is also a full Professor at Peking University, China, and an adjunct Professor at three other universities in China and the United Kingdom. He joined Peking University in 1987, where he became one of ten university-wide exceptionally promoted young Associate Professors in 1988 and further been exceptionally promoted to a full Professor in 1992. From 1989 to 1993, he worked at several universities in Finland, Canada, and the United States, including Harvard University, Cambridge, MA, and the Massachusetts Institute of Technology, Cambridge. He joined CUHK in 1993 as a Senior Lecturer and then took the current Professor position in 1996. He has published more than 200 academic papers, with a number of them being well-cited contributions. He has given a number of keynote/plenary/invited/tutorial talks in international major neural networks conferences, such as WCNN, IEEE-ICNN, IJCNN, ICONIP, etc.

Dr. Xu is on the Governor Board of the International Neural Network Society), a past president of APNNA, and an associate editor for six international journals on neural networks, including *Neural Networks*, and the IEEE TRANSACTIONS ON NEURAL NETWORKS. He was a ICONIP'96 Program Committee Chair and a General Chair of IDEAL'98, IDEAL'00. Also, he has served as Program Committee Members in international major neural networks conferences in recent years, including IJCNN (1997, 1999, 2000, 2001), WCNN (1995, 1996), IEEE-ICNN (1996), etc. He has received several Chinese national prestigious academic awards (including the National Nature Science Award) and also some international awards (including an 1995 INNS Leadership Award).