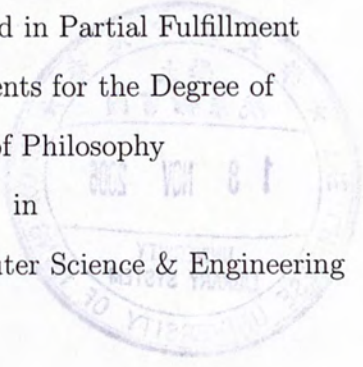


A Study on Model Selection of Binary and Non-Gaussian Factor Analysis

An, Yujia

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Department of Computer Science & Engineering



©The Chinese University of Hong Kong

January, 2005

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or the whole of the materials in this thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.



A Study on Model Selection of Binary and Non-Gaussian Factor Analysis

submitted by

An, Yujia

for the degree of Master of Philosophy
at the Chinese University of Hong Kong

Abstract

With uncorrelated Gaussian factors extended to mutually independent factors beyond Gaussian, the conventional factor analysis is extended to what is recently called independent factor analysis. Typically, it is called Binary factor analysis (BFA) when the factors are binary, and called Non-Gaussian factor analysis (NFA) when the factors are from real non-Gaussian distributions. A crucial issue in both BFA and NFA is the determination of the number of factors. In the literature of statistics, there are a number of model selection criteria that can be used for this purpose. Also, the Bayesian Ying-Yang (BYY) harmony learning provides a new principle for this purpose. This thesis presents our work on model selection for BFA and NFA. Firstly, we present our analysis on automatic model selection during BYY harmony learning for BFA. Based on such an analysis we propose two heuristic methods and a combination strategy to reduce or avoid the local minima and to obtain higher correct rate. Secondly, we investigate the BYY criterion or BYY harmony

learning with automatic model selection (BYY-AUTO) in comparison with existing typical criteria, including Akaike's information criterion (AIC), the consistent Akaike's information criterion (CAIC), the Bayesian inference criterion (BIC), and the cross-validation (CV) criterion for the model selection of BFA and NFA respectively. The study is made via experiments on data sets with different sample sizes, data space dimensions, noise variances, and hidden factors numbers. For BFA, we present the comparison of BYY criterion and BYY-AUTO with other typical model selection criteria and the experiments have shown that in most cases BIC outperforms AIC, CAIC, and CV while the BYY criterion and BYY-AUTO are either comparable with or better than BIC. Furthermore, BYY-AUTO takes much less time than the conventional two-stage learning methods with an appropriate model automatically determined during parameter learning. For NFA, the comparison of BYY criteria with other model selection criteria is performed and experiment shows that, in most cases, BYY criterion is either comparable with or better than the best of other criteria. Furthermore, the algorithm derived from BYY harmony learning takes much less time than the conventional EM algorithm since the computational complexity grows exponentially with the number of factors in EM algorithm. Therefore, BYY harmony learning is a more preferred tool for the model selection in BFA and NFA.

摘要

Binary factor analysis (BFA) 和 non-Gaussian factor analysis (NFA) 是两种著名的多元数据分析技术。在这两种技术手段中, 一个共同的关键问题就是决定 hidden factor 的个数问题, 也就是模型选择的问题。我的论文阐述了我们在这两种多元数据分析技术中模型选择问题的研究。首先, 我们分析了利用 BYY 学习方法进行 BFA 的自动模型选择问题。在这些分析的基础上, 我们提出了两种启发性的方法和一种合成机制来减少算法中遇到的局部极小值问题, 并且利用这些方法可以得到比较高的正确率。接下来, 我们对现有的一些典型模型选择准则和 BYY 模型选择方法分别在 BFA 和 NFA 这两种技术中作了比较研究。研究是通过在各种不同的环境条件下进行实验时得出的。对 BFA 来讲, 我们比较了 BYY 准则以及 BYY 自动模型选择与其他典型的模型选择方法, 包括 AIC, CAIC, BIC 和 CV。实验证明在大多数条件下, BIC 的结果要优于其它几种典型的模型选择方法, 而 BYY 准则和 BYY 自动模型选择在大多数情况下的结果是等于或者优于 BIC 准则的。而且, BYY 自动模型选择所需要的时间远远少于其他两步法的模型选择标准。对于 NFA, 我们比较了 BYY 准则同现有的其他几种准则, 结果同 BFA 类似, 而且, BYY 所需要的时间要远少于传统最大释然算法需要的时间。因此, BYY 学习方法对于 BFA 和 NFA 而言是一种更好的选择。

Acknowledgment

I would like to take this opportunity to express my gratitude to my supervisor, Prof. Lei Xu, for his generous guidance and patience to me during my M.Phil study. I am so lucky to be supervised by Prof. Xu, from whom I learned much in many ways. Particularly, I not only learned how to do the research in a motivated way, but also understood the importance of the basic concept and physical meaning in the research process.

I thank the members of my thesis committee, Prof. Laiwan Chan, Prof. Dityan Yeung and Prof. Jun Wang, for squeezing much time on reading and commenting on my thesis.

I would also like to show my gratitude to the Department of Computer Science and Engineering, CUHK, for the provision of the best equipment and pleasant office environment for high quality research.

I want to give my thanks to my fellowship colleagues, Dr. Zhi Yong Liu, Dr. Kai Chun Chiu, and Ms. Xuelei Hu, whom give me useful help on both research and other activities and with whom I shared my happiness and depression during my Mphil period. I thank all the members in the computer science and engineering department and would like to thank in particular Mr. Calvin Tsang, Ms. Ivy Kwok, Ms. Temmy So, Ms. Cynthia, Ms. Yik Siu Yee and Ms. Au Din Zee for their kindly assistance during the past two years. Sincere thanks also go to Mr. Shi Lu, Mr. Chu Hong Hoi, Ms. Yang Lu, Dr. Xinwen Hou, and Dr. Changyin Sun.

My special thanks must go to my family and my friends who have given me the greatest support and encouragement, so that I can keep concentrated

on my postgraduate study.

Contents

Abstract	iv
Acknowledgement	vi
1 Introduction	1
1.1 Background	1
1.1.1 Review on BFA	2
1.1.2 Review on CPA	2
1.1.3 Typical model selection criteria	3
1.1.4 New molecular chain algorithm and model selection criteria	3
1.1.5 Summary	3
1.2 Our work between BFA and CPA	4
1.3 Terminology	5
2 Construction of P and BI architectures for BFA with genetic matrix model selection	6
2.1 Improvement of BFA using HYT model selection criteria towards model selection	6
2.1.1 Structure of PFA	6

Contents

Abstract	ii
Acknowledgement	iv
1 Introduction	1
1.1 Background	1
1.1.1 Review on BFA	2
1.1.2 Review on NFA	3
1.1.3 Typical model selection criteria	5
1.1.4 New model selection criterion and automatic model selection	6
1.2 Our contributions	7
1.3 Thesis outline	8
2 Combination of B and BI architectures for BFA with automatic model selection	10
2.1 Implementation of BFA using BYY harmony learning with automatic model selection	11
2.1.1 Basic issues of BFA	11

2.1.2	B-architecture for BFA with automatic model selection	12
2.1.3	BI-architecture for BFA with automatic model selection	14
2.2	Local minima in B-architecture and BI-architecture	16
2.2.1	Local minima in B-architecture	16
2.2.2	One unstable result in BI-architecture	21
2.3	Combination of B- and BI-architecture for BFA with automatic model selection	23
2.3.1	Combine B-architecture and BI-architecture	23
2.3.2	Limitations of BI-architecture	24
2.4	Experiments	25
2.4.1	Frequency of local minima occurring in B-architecture	25
2.4.2	Performance comparison for several methods in B-architecture	26
2.4.3	Comparison of local minima in B-architecture and BI- architecture	26
2.4.4	Frequency of unstable cases occurring in BI-architecture	27
2.4.5	Comparison of performance of three strategies	27
2.4.6	Limitations of BI-architecture	28
2.5	Summary	29

3 A Comparative Investigation on Model Selection in Binary

Factor Analysis	31
3.1 Binary Factor Analysis and ML Learning	32
3.2 Hidden Factors Number Determination	33
3.2.1 Using Typical Model Selection Criteria	33
3.2.2 Using BYY harmony Learning	34

3.3	Empirical Comparative Studies	36
3.3.1	Effects of Sample Size	37
3.3.2	Effects of Data Dimension	37
3.3.3	Effects of Noise Variance	39
3.3.4	Effects of hidden factor number	43
3.3.5	Computing Costs	43
3.4	Summary	46
4	A Comparative Investigation on Model Selection in Non-gaussian Factor Analysis	47
4.1	Non-Gaussian Factor Analysis and ML Learning	48
4.2	Hidden Factor Determination	51
4.2.1	Using typical model selection criteria	51
4.2.2	BYY harmony Learning	52
4.3	Empirical Comparative Studies	55
4.3.1	Effects of Sample Size on Model Selection Criteria	56
4.3.2	Effects of Data Dimension on Model Selection Criteria	60
4.3.3	Effects of Noise Variance on Model Selection Criteria	64
4.3.4	Discussion on Computational Cost	64
4.4	Summary	68
5	Conclusions	69
	Bibliography	71

List of Tables

2.1	Un-global result for B-architecture.	17
2.2	Unstable result in BI-architecture.	22
2.3	Rates of the special local minima (SL), other local (OL) and correct cases (C) in B-architecture in 100 experiments	25
2.4	Rates of the special local minima (SL), other local (OL) and correct cases(C) in comparison of different methods by 100 experiments	26
2.5	Rates of the local minima in original B-architecture and BI-architecture in 100 experiments	27
2.6	Rates of unstable cases (US), local minima (L) and correct cases (C) in BI-architecture by 100 experiments	27
2.7	Performance comparison of the three strategies, B-architecture, BI-architecture, and the combination strategy	28
2.8	Average results as σ_0^2 increases	29
2.9	Average results as the mean value of x increases.	29

3.1	Rates of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different sample sizes for BFA in 100 experiments	39
3.2	Rates of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different data dimensions for BFA in 100 experiments	41
3.3	Rates of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different noise variances for BFA in 100 experiments	43
3.4	Rates of underestimating (U), success (S), and overestimating (O) by each criterion on simulation data sets with different hidden factors numbers for BFA in 100 experiments	44
3.5	CPU times on the simulation data sets with $n = 100$, $d = 9$, and $k = 3$ for BFABy using the EM algorithm for AIC, BIC, CAIC and CV, algorithm (3.10) for BYY criterion and (2.6) for BYY-AUTO	45
3.6	CPU times on the simulation data sets with $n = 100$, $d = 9$, and $k = 3$ for BFA by using the EM algorithm and the algorithm derived from BYY harmony learning where set the candidate $k = 3$	45
4.1	Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different sample sizes for selecting hidden factors number k with $k_j = 3$ fixed for NFA in 50 experiments	57

4.2	Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different sample sizes for selecting gaussians number k_j with $k = 3$ fixed for NFA in 50 experiments	60
4.3	Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different data dimensions for selecting hidden factors number k with $k_j = 3$ fixed for NFA in 50 experiments	61
4.4	Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different data dimensions for selecting gaussians number k_j with $k = 3$ fixed for NFA in 50 experiments	61
4.5	Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different noise variances for selecting hidden factors number k with fixing $k_j = 3$ for NFA in 50 experiments	65
4.6	Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different noise variances for selecting gaussians number k_j with fixing $k = 3$ for NFA in 50 experiments	65
4.7	CPU time results on the simulation data sets with $n = 40$, $d = 7$, and $k = 3$ for NFA by using the EM algorithm for AIC, BIC, CAIC and CV, algorithm Eq. 4.22 for BYY criterion . . .	67

- 4.8 CPU time results on the simulation data sets with $n = 40$, $d = 7$, and $k = 3$ for NFA by using the EM algorithm and the algorithm derived from BYY harmony learning where set the candidate $k = 3$ 68

List of Figures

- 3.1 The curves obtained by the criteria AIC, BIC, CAIC, GIC, CV and BYY on the data sets of a 3-dimensional p ($p = 3$) generated from a 3-dimensional p ($k = 3$) with independent noise for NFA 41
- 3.2 The curves obtained by the criteria AIC, BIC, CAIC, GIC, CV and BYY on the data sets of a 3 with the all dependent generated from a 3-dimensional p ($k = 3$) for NFA 42
- 3.3 The curves obtained by the criteria AIC, BIC, CAIC, GIC, CV and BYY on the data sets of a 3-dimensional p ($p = 3$) generated from a 3-dimensional p ($k = 3$) with dependent noise for NFA 43
- 3.4 The curves obtained by the criteria AIC, BIC, CAIC, GIC, CV and BYY for learning the factors on data sets generated from a 7-dimensional p ($p = 7$) generated from a 3-dimensional p ($k = 3$) with different correlation for NFA 44

List of Figures

3.1	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY on the data sets of a 9-dimensional x ($d = 9$) generated from a 3-dimensional y ($k = 3$) with different sample sizes for BFA.	38
3.2	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY on the data sets of a x with different dimensions generated from a 3-dimensional y ($k = 3$) for BFA.	40
3.3	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY on the data sets of a 9-dimensional x ($d = 9$) generated from a 3-dimensional y ($k = 3$) with different noise variances for BFA.	42
4.1	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the factors number k with $k_j = 3$ fixed on the data sets of a 7-dimensional x ($d = 7$) generated from a 3-dimensional y ($k = 3$) with different sample sizes for NFA.	58

4.2	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the gaussians number k_j with $k = 3$ fixed on the data sets of a 7-dimensional x ($d = 7$) generated from a 3-dimensional y ($k = 3$) with different sample sizes for NFA.	59
4.3	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the factors number k with fixing $k_j = 3$ on the data sets of a x with different dimensions generated from a 3-dimensional y ($k = 3$) for NFA.	62
4.4	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the gaussians number k_j with fixing $k = 3$ on the data sets of a x with different dimensions generated from a 3-dimensional y ($k = 3$) for NFA.	63
4.5	The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the factors number k and the gaussians number k_j on the data sets of a 7-dimensional x ($d = 7$) generated from a 3-dimensional y ($k = 3$) with different noise variances for NFA.	66

Chapter 1

Introduction

1.1 Background

Factor analysis (FA) is a well-known multivariate analysis technique in help of the following linear model [3, 39]

$$x = Ay + c + e, \tag{1.1}$$

where x is a d -dimensional random vector of observable variables, e is a d -dimensional random vector of unobservable noise variables and is drawn from Gaussian, A is a $d \times k$ loading matrix, c is a d -dimensional mean vector and y is a k -dimensional random vector of unobservable mutually uncorrelated Gaussian factors. y and e are mutually independent. However, FA is not appropriately applicable to the real world data that can not be described as generated from Gaussian factors. With uncorrelated Gaussian factors extended to mutually independent factors beyond Gaussian, FA is extended to what is recently called independent factor analysis. Typically, it is called Binary factor analysis (BFA) when the factors are binary, and called Non-Gaussian factor analysis (NFA) when the factors are from real non-Gaussian distributions.

One other crucial issue in implementing BFA and NFA is appropriately determining the number of hidden factors, i.e., the dimension of y in Eq. 1.1, which is a typical model selection problem. Conventionally, it needs a two-phase style implementation that first conducts parameter learning on a set of candidate models under the maximum likelihood (ML) principle and then select the ‘optimal’ model among the candidates according to a given model selection criterion. Such conventional model selection methodology is computationally inefficient because it involves enumeration of the objective function for different candidate models. In contrast, model selection can be automatically done during parameter estimation in Bayesian Ying Yang (BYY) harmony learning. Moreover, a model selection criterion has been also proposed based on BYY harmony learning. Among these model selection criteria and methods, which ones are suitable for binary factor analysis (BFA) as well as non-Gaussian factor analysis (NFA)?

1.1.1 Review on BFA

Being different from the conventional factor analysis where factors are assumed to be Gaussian, binary factor analysis (BFA) regards the observable variables as generated from a small number of latent binary factors. In practice, BFA has been widely used in various fields, especially in the social science such as political science, educational testing, psychological measurement as well as the tests of disease severity, etc [28]. Also, it can be used for data reduction, data exploration, or theory confirmation [39].

Many studies have been made on BFA in literature. One direction includes studies under the names of latent trait model (LTA) and latent class model, both are called by a joint name latent structure model. The basic issue of them is to fit a latent trait or class model under the Item Response Theory (IRT) approach, which is the study of test and item scores based on assumptions concerning the mathematical relationship between abilities and item responses [5, 16]. Another typical example is the multiple cause model that considers the observed samples generated from independent binary hidden factors, trained by either cost minimization or maximum likelihood. [24, 15]. It differs from other models by permitting clusters not only to compete for data points, but also to cooperate with one another in accounting for observed data [23]. One other example is the auto-association network that is trained by back-propagation via simply copying input as the desired output [9, 8]. Recently, one new model for BFA is proposed by Xu with particular and attractive feature about model selection [36, 38, 39]. That is, BYY harmony learning with one hidden layer of binary units (BHL-BYY). It can make model selection implemented either *automatically* during parameter learning or *subsequently* after parameter learning via a new class of model selection criteria [38, 39].

1.1.2 Review on NFA

In the real world, data are usually generated from non-Gaussian factors. In such cases, conventional factor analysis is inappropriate. Non-Gaussian factor analysis (NFA) was proposed by Xu in 1998 which generalizes the conventional

FA by assuming that each hidden factor follows non-Gaussian distribution [33]. NFA not only avoids the rotation and additive indeterminacies encountered by classical FA, but also relaxes the impractical noise-free assumption for independent component analysis (ICA) [42]. In fact, NFA can handle both noiseless mixing problem and the general case where the number of observables differs from the number of sources and the data are noisy [4]. Therefore, NFA is a very promising tool for the well-known blind source separation (BSS) problem, dependence reduction, and structure discovery problem [4, 18].

In recent years, studies related to NFA have been carried out. One kind of examples consists of the efforts under the name noisy ICA. One example is given in [17] where a so-called joint maximum likelihood is considered to be maximized. However, a rough approximation is actually used there and also how to specify a scale remains an open problem yet [38]. Some other noisy ICA examples are also referred to [18, 14]. In [20], an approach that exactly implements ML learning for the model Eq. (1.1) was firstly proposed. Similar to [31, 32], they considered modelling each non-Gaussian factor by a Gaussian mixture. In help of a trick that the product of summations is equivalently exchanged into a summation of products, the integral in computing likelihood becomes a summation of a large number of analytically computable integrals on Gaussians, which makes an exact ML learning on Eq. (1.1) implemented by an exact EM algorithm. The same approach has been also published in [4] under the name of independent factor analysis. However, the number of terms of computable integrals on Gaussians grows exponentially with the number of factors, and it correspondingly incurs exponentially growing computing costs. In contrast, computing costs of implementing the NFA algorithms in [36, 38]

grow only linearly with the number of factors, as demonstrated in [40, 42].

1.1.3 Typical model selection criteria

Determination of the hidden factors number is a crucial model selection problem in the implementation of BFA and NFA. This determination can be achieved via several existing statistical model selection criteria. However, in practice most studies still assume a known model scale or determine it heuristically. One main reason is that these criteria have to be implemented in a two-phase procedure that is very time-consuming. First, we need to assume a range of values of the model scale K from K_{min} to K_{max} which is assumed to contain the optimal K . At each specific model scale K , we estimate the parameters θ under the ML learning principle. Second, we make the following selection

$$\hat{K} = \arg \min_K \{J(\hat{\theta}, K), K = K_{min}, \dots, K_{max}\}, \quad (1.2)$$

where $J(\hat{\theta}, K)$ is a given model selection criterion.

Three typical model selection criteria are the Akaike's information criterion (AIC) [1, 2], its extension called Bozdogan's consistent Akaike's information criterion (CAIC) [10], and Schwarz's Bayesian inference criterion (BIC) [25] which coincides with Rissanen's minimum description length (MDL) criterion [21, 6]. These three model selection criteria can be summarized into the following general form [26]

$$J(\hat{\theta}, K) = -2L(\hat{\theta}) + B(n)Q(K) \quad (1.3)$$

where $L(\hat{\theta})$ is the log likelihood based on the ML estimation $\hat{\theta}$ under a given K , and $Q(K)$ is the number of free parameters in K -scale model. Moreover,

$B(n)$ is a function with respect to the number of observations n as follows:

- $B(n) = 2$ for Akaike's information criterion (AIC) [2, 1],
- $B(n) = \ln(n) + 1$ for Bozdogan's consistent Akaike's information criterion (CAIC) [10],
- $B(n) = \ln(n)$ for Schwarz's Bayesian inference criterion (BIC) [25].

Another well-known model selection technique is cross-validation (CV). By this technique data are repeatedly partitioned into two sets, one is used to build the model and the other is used to evaluate the statistic of interest [27]. For the i th partition, let U_i be the data subset used for testing and U_{-i} be the remainder of the data used for training, the cross-validated log-likelihood for a K -scale model is

$$J(\hat{\theta}, K) = -\frac{1}{m} \sum_{i=1}^m L(\hat{\theta}(U_{-i})|U_i) \quad (1.4)$$

where m is the number of partitions, $\hat{\theta}(U_{-i})$ denotes the ML parameter estimations of K -scale model from the i th training subset, and $L(\hat{\theta}(U_{-i})|U_i)$ is the log-likelihood evaluated on the data set U_i . Featured by m , it is usually referred as making an m -fold cross-validation or shortly m -fold CV.

1.1.4 New model selection criterion and automatic model selection

Bayesian Ying-Yang (BYY) harmony learning was proposed as a unified statistical learning framework firstly in [30] and systematically developed in past

years. From the perspective of general learning framework, the BYY harmony learning consists of a general BYY system and a fundamental harmony learning principle as a unified guide for developing new regularization techniques, a new class of criteria for model selection, and a new family of algorithms that perform parameter learning with automated model selection. The BYY learning with specific structures applies to unsupervised learning, supervised learning, and state space approach for temporal modelling, with a number of new results. The details are referred to [35, 36, 37, 38].

By applying BYY learning to BFA and NFA respectively, not only new criteria for model selection of BFA and NFA are obtained, but also adaptive algorithms are developed that perform BFA and NFA with an appropriate number of hidden factors automatically determined during adaptive learning [36].

1.2 Our contributions

This section briefly summarizes the unique aspects and important contributions in this thesis, which fall in the following three main areas.

1. We analyze the experiment results of two architectures of BYY harmony learning with automatic model selection during parameter estimation in implementation on BFA. Based on the analysis, we proposed two heuristic methods and a combination strategy which can efficiently improve the correct rate for implementing BFA with model selection done automatically.

2. We investigate the BYY model selection criterion and automatic model selection ability on the implementation of BFA, compared with the criteria of AIC, CAIC, BIC, and CV in various situations. Experiment results have shown that the performance of BIC is superior to AIC, CAIC, and CV in most cases. The BYY criterion and the BYY automatic model selection learning (BYY-AUTO) are, in most cases, comparable with or even superior to the best among of BIC, AIC, CAIC, and CV. Moreover, BYY-AUTO takes much less time than the conventional two-phase model selection methods. Thus, BYY learning is a more preferred tool for BFA.

3. We make a comparison of the BYY model selection criterion and the criteria of AIC, CAIC, BIC, and CV on the implementation of NFA in various situations. Experiment results have shown that in most cases, the BYY criterion is comparable with or even superior to the best among AIC, BIC, CAIC, and CV. Moreover, experiments have also shown that the algorithm derived from BYY harmony learning converges faster than the EM algorithm of ML learning for the criteria BIC, AIC, CAIC, and CV.

1.3 Thesis outline

The thesis is organized as follows:

Chapter 2 is devoted to analyze the experiment results of two architectures

for BYY harmony learning, B-architecture and BI-architecture, on the implementation of BFA with automatic model selection. Based on the analysis, two heuristic methods are proposed for B-architecture to improve the correct rate. Furthermore, a combination strategy of B-and BI-architecture is proposed to achieve the much higher correct rate. Experiments have been done to show the performance of these proposed methods.

Chapter 3 describes the study on the BYY criterion and BYY harmony learning with automatic model selection (BYY-AUTO) in comparison with existing typical criteria AIC, BIC, CAIC, and CV for the model selection of binary factor analysis. Comparative experiments are given in this chapter under various situations which have shown that the BYY criterion and the BYY automatic model selection learning (BYY-AUTO) are, in most cases, comparable with or even superior to the best among of BIC, AIC, CAIC, and CV. The comparison of computational cost shows that BYY-AUTO takes much less time than all the other two-stage model selection criteria.

Chapter 4 describes the investigation on the BYY criterion in comparison with the typical criteria AIC, BIC, CAIC, and CV for the model selection of non-Gaussian factor analysis. We present the experimental comparison in this chapter to show that in most cases BYY criterion is comparable or better than the other criteria and the running time for BYY criterion based on BYY harmony learning is less than the other criteria based on the EM algorithm.

Finally, Chapter 5 concludes the whole thesis and discusses our future work.

Chapter 2

Combination of B and BI architectures for BFA with automatic model selection

In this chapter, we concentrate on automatic model selection ability of BYY harmony learning for implementing BFA. We mainly analyze the experiment results for B-architecture and BI-architecture, two typical architectures in BYY harmony learning. Based on the analysis, two heuristic methods to avoid local minima for B-architecture are proposed. Moreover, based on the analysis and comparison of these two architectures, we propose a combination strategy of them for making use of both advantages of them and getting high correct rate.

2.1 Implementation of BFA using BYY harmony learning with automatic model selection

2.1.1 Basic issues of BFA

The model Eq. 1.1 is called Binary Factor Analysis (BFA) when y is a binary random vector. Since the Bernoulli distribution is a most familiar binary distribution and the BYY harmony learning for BFA in [39] assumes y a Bernoulli, in our work we also assume that y comes from the following multivariate Bernoulli distribution: [36, 39]

$$p(y) = \prod_{j=1}^k [q_j \delta(y^{(j)}) + (1 - q_j) \delta(1 - y^{(j)})], \quad (2.1)$$

where q_j is the probability that $y^{(j)}$ takes the value 1. In this paper, we consider that e is from a spherical Gaussian distribution for the binary factor analysis model, i.e., $p(x|y)$ has the following form:

$$p(x|y) = G(x|Ay + c, \sigma^2 I) \text{ (for BFA)}, \quad (2.2)$$

where $G(x|Ay + c, \sigma^2 I)$ denotes a multivariate normal (Gaussian) distribution with mean $Ay + c$ and spherical covariance matrix $\sigma^2 I$, with I being a $d \times d$ identity matrix.

2.1.2 B-architecture for BFA with automatic model selection

Bayesian Ying-Yang harmony learning consider the world consists of a observation space x and the corresponding inner representation space y . On one hand, the observation x can be regarded as generated from the inner representation y via a backward path distribution $p(x|y)$. On the other hand, we can interpret that each x is mapped into an invisible inner representation y via a forward path distribution $p(y|x)$. Since the BFA model as Eq. 1.1 mainly focuses on the generative direction from inner representation domain to the observation domain, the following two architectures covering the generative direction in BYY harmony learning can be adopted for the implementation of BFA:

1. Backward architecture (B-architecture): $p(y|x)$ is structure-free and $p(x|y)$ is parametric.
2. Bidirectional architecture (BI-architecture): Both $p(y|x)$, $p(x|y)$ are parametric.

In B-architecture, the structure of $p(x|y)$ is pre-defined as Eq. 2.2 while $p(y|x)$ is structure-free and is indirectly specified by the structures of $p(y)$ and $p(x|y)$. With automatic determination of model scale which is the hidden binary factors number k in BFA, the parameter estimation and the determination of k share a same harmony function. So, these two subtasks can be implemented together via BYY harmony learning, i.e.,

$$(\theta, k) = \arg \max_{\theta, k} H(\theta, k), \quad (2.3)$$

where the parameters θ need to be learned include the factor loading A , the mean vector c , the variance σ^2 , and the probability q_j for each factor $y^{(j)}$.

According to Sect.4.1 in [39], especially Eq. 21, Eq. 30, Eq. 31 and Eq. 32, the specific form of $H(\theta, k)$ is given as following:

$$H(\theta, k) = -\frac{d}{2} \ln \sigma^2 - \frac{1}{2n} \sum_{t=1}^n \frac{\|x_t - Ay_t - c\|^2}{\sigma^2} + \frac{1}{n} \sum_{t=1}^n \sum_{j=1}^k [y_t^{(j)} \ln q_j + (1 - y_t^{(j)}) \ln(1 - q_j)], \quad (2.4)$$

where $y_t^{(j)}$ is the j -th element in vector y_t , and

$$y_t = \arg \max_y H(\theta, k). \quad (2.5)$$

This harmony function can be regarded as two parts. Maximizing it means minimizing the front part $\frac{d}{2} \ln \sigma^2 + \frac{1}{2n} \sum_{t=1}^n \frac{\|x_t - Ay_t - c\|^2}{\sigma^2}$ and maximizing the latter part $\frac{1}{n} \sum_{t=1}^n \sum_{j=1}^k [y_t^{(j)} \ln q_j + (1 - y_t^{(j)}) \ln(1 - q_j)]$ synchronously. The action on the latter part will lead to the fact that q_j of each redundant factor $y^{(j)}$ is pushed towards 0 or 1, thus we can discard the corresponding redundant factors [36, 38, 39]. Setting k large enough initially (e.g., the dimensionality of x in this paper), we can implement the adaptive algorithm with model selection performed automatically. Shortly, we refer it by BYY automatic model selection learning (BYY-AUTO).

Following the procedure given in Table 1 of [39] or the algorithm by Eq. 45 in [41], we implement BYY-AUTO by the following adaptive algorithm:

- step 1: get y_t by Eq. 2.5,
- step 2: (a) update
 - $e_t = x_t - Ay_t - c,$

$$\sigma^{2 \text{ new}} = (1 - \eta)\sigma^{2 \text{ old}} + \eta \text{tr}[e_t e_t^T]/d,$$

$$A^{\text{new}} = A^{\text{old}} + \eta e_t y_t^T,$$

$$c^{\text{new}} = c^{\text{old}} + \eta e_t,$$

where η is a step constant,

(b) update $q_j^{\text{new}} = b_j^{\text{new} 2}$ by

$$b_j^{\text{new}} = b_j + \eta b_j (y_t^{(j)} - q_j) / [q_j (1 - q_j)],$$

if either $q_j^{\text{new}} \rightarrow 0$ or $q_j^{\text{new}} \rightarrow 1$,

discard the j th dimension of y . (2.6)

where η is the step length constant.

2.1.3 BI-architecture for BFA with automatic model selection

For BI-architecture, except for $p(x|y)$, the structure of $p(y|x)$ also needs to be prefixed. We introduce a specific sigmoid function as [39]

$$p(y|x) = \delta(y - y(x)), y(x) = s(\hat{y}), \hat{y} = Wx + v. \quad (2.7)$$

where and hereafter in this thesis, $s(r)$ denotes a sigmoid function $s(r) = (1 + e^{-\beta r})^{-1}$.

As given in Sect. 4.2 of [39], the BI-architecture based adaptive algorithm for implementing BFA in [39] is written as following:

step 1: $y_t = s(z_t), z_t = Wx_t + v,$

step 2: (a) update A, c, σ^2 as step 2 (a) in Eq. 2.6,

(b) update q_j as step 2 (b) in Eq. 2.6

$$\begin{aligned}
& \text{if either } q_j^{\text{new}} \rightarrow 0 \text{ or } q_j^{\text{new}} \rightarrow 1, \\
& \text{discard the } j\text{th dimension of } y, \\
\text{(c) update} \\
& e_t^0 = A^{\text{old}T} e_t \\
& v^{\text{new}} = v^{\text{old}} + \eta D_s(z_t) e_t^0, \\
& W^{\text{new}} = W^{\text{old}} + \eta D_s(z_t) e_t^0 x_t^T, \tag{2.8}
\end{aligned}$$

where the notation $D_s(u)$ denotes a diagonal matrix with its diagonal elements $[s'(u^{(1)}), \dots, s'(u^{(k)})]^T$, $s'(r) = ds(r)/dr$ and $s(r)$ denotes a sigmoid function.

B-architecture is quite computational expensive since the process of getting y_t by Eq. 2.5 needs to try all the possible value of y . Moreover, the process of maximizing the cost function $H(\theta, k)$ in Eq. 2.4 causes the local minima problem which will be discussed in the next subsection. In contrast, the computational cost can be efficiently saved in BI-architecture by considering $p(y|x)$ in an appropriate parametric structure, such as the special case in Eq. 2.7 that we used in our work. Furthermore, experiments have shown that the local minima caused by maximizing $H(\theta, k)$ can be considerably reduced in BI-architecture. In the next section, based on the analysis of B- and BI-architecture a new strategy of combining these two architectures will be proposed.

2.2 Local minima in B-architecture and BI-architecture

This section mainly analyzes the experiment results of B- and BI-architecture with automatic model selection during parameter learning in BFA.

2.2.1 Local minima in B-architecture

One frequent case in B-architecture

After a large number of experiments, we found that in B-architecture an exceptional but interesting result often appears especially when the dimensionality of x is very large. That is, there are one or more than one redundant q_j s can't be pushed to 0 or 1 and the corresponding redundant hidden factors cannot be discarded automatically. But each of them is learned to be the same as some correct hidden factor. In another word, more than one factors are learned to be the same.

We give an example to illustrate this case. A data set includes 100 samples generated from a hidden binary y with 5 hidden factors (or components). The dimensionality of data x is 10, so the true loading matrix is a 10×5 matrix in the generative function. We apply this data set to the BFA model using B-architecture with the initial value of the hidden factors number k equal to 10. There are 5 redundant components in y . Theoretically, the corresponding probabilities q_j s of the 5 redundant components will be pushed towards 0 or 1 during learning and then be discarded. But the experiment result contains 6 components which means one redundant q_j can't be pushed to 0 or 1 and

all the five correctly learned components are contained in the result. Very interestingly, this redundant component is learned to be the same as some correct component such as $y^{(9)} = y^{(1)}$ and the corresponding columns of these two components in the loading matrix A are linear relevant.

For clarity we idealize and abstract this result to the following Table. 2.1, where there is only one extra component that is y_{re} and k is the true number of hidden factors. In this table, \vec{a}_i means the i -th column of the loading matrix A and all the variables with the subscript 0 such as c_0 and σ_0^2 denote the true ones in the original generated function.

Table 2.1: Un-global result for B-architecture.

Para.	Features
x	$x = \left(\vec{a}_1 \quad \dots \quad \vec{a}_k \quad \vec{a}_{re} \right) \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(k)} \\ y^{(re)} \end{pmatrix} + \vec{c} + e$
A	$\begin{cases} \vec{a}_{ex} + \vec{a}_1 = \vec{a}_{01} \\ \vec{a}_i = \vec{a}_{0i} \end{cases} \quad \text{for } i = 2 \dots k$
y	$\begin{cases} y^{(ex)} = y^{(1)} = y_0^{(1)} \\ y^{(i)} = y_0^{(i)} \end{cases} \quad \text{for } i = 2 \dots k$
c	$c = c_0$
σ^2	$\sigma^2 = \sigma_0^2$
q_j	$\begin{cases} q_{re} = q_1 = q_{01} \\ q_i = q_{0i} \end{cases} \quad \text{for } i = 2 \dots k$

Although this case breaches the fact that the hidden factors are independent mutually, it often happens in experiment results of B-architecture. Table. 2.3 in sect. 2.4.1 shows that the frequency is 45% of appearing this case in B-architecture.

After calculating the mean square error for \mathbf{x} and \mathbf{y} and checking out the first and second order differentials, we believe that the result is one kind of local minima. In conclusion, the local case described in this section often appears in the results of a B-architecture in implementing BFA. Although there are many other local minima, a large number of experiments have shown that this special case is a most frequent local minima in B-architecture.

Two heuristic methods to reduce the local minima in B-architecture

In the above section, we have presented a case of the local minima that often appears in a B-architecture. In this section, we present two heuristic methods to reduce these local minima.

After analyzing, we found that the learning results are highly affected by the initial learning rate of parameter A , denoted by η_{0_A} . If η_{0_A} is too small, the learning results will easily fall into the local minima. Suppose the data are actually generated from k hidden factors. To obtain the correct rudiment of A_0 relies not only on the initialization of A but also on the initial step length η_{0_A} . If this value is properly big enough, A will be learned to rapidly obtain the rudiment of A_0 within 5-10 iterations. After several iterations, the learning rate of A can be reduced to a smaller constant or reduced step by step. We give an intuitive explanation for the effect of this method. When η_{0_A} is big

enough, the value of A can jump acutely in each iteration and it can rapidly find the global optimal value in a large range. In other words, it can reduce the probability of falling into a local minimum for B-architecture.

The proper initial learning rate of A is usually in the range of 0.1 to 0.6 experimentally. One feasible approach in practice is to learn several times with the initial learning rate η_{0_A} changing in some range such as 0.1 – 0.5 and to find the best result. In the section of experiment, a comparison of the probabilities which the local minima occur using this heuristic method or not is given in Table. 2.4 of sect. 2.4.2. From the statistics, it can be found that the frequency of falling into local points has been obviously decreased when this heuristic method is used.

Based on an analysis of the first heuristic method, we have another trick. That is, setting the initial values of parameter A so that the difference between its columns are distinct. It can avoid some kinds of local minimum. But in this case, η_{0_A} still needs to be large enough. So the combination of these two efforts will be better to reduce local minima for B-architecture.

Improved implementation for B-architecture with automatic model selection

One key feature of the local minima described in the above section is that the independent condition of the hidden space y is broken. Correspondingly, the factor loading A has not full rank. In fact, we can remove this kind of local minima via making A non-singularity [41]. In implementation, we make the eigenvalue decomposition of A for each iteration. If A has not full rank, we

can discard the 0 eigenvalue column of A and the corresponding hidden factors of in y to guarantee A non-singularity. An improved adaptive algorithm for B-architecture is given in the following equation [41]:

$$\begin{aligned}
 \text{step 1:} & \quad \text{get } y_t \text{ by Eq. 2.5,} \\
 \text{step 2: (a)} & \quad \text{update } A, c, \sigma^2 \text{ as step 2 (a) in Eq. 2.6,} \\
 & \quad \text{(b) update } q_j \text{ as step 2 (b) in Eq. 2.6} \\
 & \quad \text{if either } q_j^{\text{new}} \rightarrow 0 \text{ or } q_j^{\text{new}} \rightarrow 1, \\
 & \quad \text{discard the } j\text{th dimension of } y, \\
 & \quad \text{(c) make eigenvalue decomposition } A = U\Lambda V'. \\
 & \quad \text{If } \Lambda \text{ is not full rank, make linear transform} \\
 & \quad \text{to } A, \text{ then delete 0 value columns of } A \text{ and} \\
 & \quad \text{the corresponding dimensions of } y \text{ to} \\
 & \quad \text{guarantee } A \text{ has full rank.} \tag{2.9}
 \end{aligned}$$

where η is the step length constant. The above step (c) checks the satisfaction of equation (40) in [41] for a full rank independent space. To save computing cost, this can also be made via updating U, Λ, V as suggested by equation (470) in [41].

Although this improved algorithm can remove the special local minima discussed in this section, there still are some other local minima in the result of B-architecture, which can be found in Table. 2.4. These local minima still have effect to the result of B-architecture.

2.2.2 One unstable result in BI-architecture

Because of introducing a sigmoid function which is highly non-linear in BI-architecture, most of the local minima can be avoided in BI-architecture for the implementation of BFA with automatic model selection [39, 29]. Experiment in sect. 2.4.3 has confirmed this conclusion. Even so, the experimental results still show some blemishes for BI-architecture. Via experiments it was found that there are many unstable cases in the results of BI-architecture even if the iteration time is very large. If such case occurs, there must be some components that cannot converge. There are some distinct features in such unstable results. We present one representative form of this case as in Table. 2.2.

The unstable case illustrated in Table. 2.2 is very difficult to converge in practical implementation because the redundant factor y_{re} can change within the two binary numbers 0 and 1 at each learning iteration and probability q_{re} can also change within the domain from 0 to 1 correspondingly. The unstable result described in sect.2.2.2 often occurs in BI-architecture especially when the number of hidden factors increases. Table. 2.6 in sect. 2.4.4 shows that the frequency of occurring this case in BI-architecture is high to 33%.

We have made some analysis about this unstable case. As we have discussed in sect.2.1.2, B-architecture achieves automatic model selection via maximizing the harmony function Eq. 2.4 to push the probabilities q_j towards 0 or 1 in most degree. Moreover, the method to get y_t in each iteration is also a optimal process and more accurate than BI-architecture. Therefore, if this unstable case occurs in B-architecture, it will rapidly convergent to the result that the unstable redundant q_{re} is pushed to 0 or 1 and then is discarded. In contrast,

Table 2.2: Unstable result in BI-architecture.

Para.	Features
x	$x = \left(\vec{a}_1 \quad \dots \quad \vec{a}_k \quad \vec{a}_{re} \right) \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(k)} \\ y^{(re)} \end{pmatrix} + \vec{c} + e$
A	$\begin{cases} \vec{a}_i = \vec{a}_{0_i}, \text{ for } i = 1 \dots k \\ \vec{a}_{re} = 0 \end{cases}$
y	$\begin{cases} y^{(i)} = y_0^{(i)}, \text{ for } j = 1 \dots k \\ y^{(re)} = \text{arbitraty binary number in 0 and 1} \end{cases}$
c	$c = c_0$
σ^2	$\sigma^2 = \sigma_0^2$
q_j	$\begin{cases} q_i = q_{0_i}, \text{ for } i = 1 \dots k \\ q_{ex} = \text{Unstable number between 0 and 1} \end{cases}$

for BI-architecture there is no such powerful strength and the function is highly nonlinear. The value of q_j only relies on the learning result of $y^{(j)}$, and q_j can be learned towards 0 or 1 only when $y^{(j)}$ is pushed to 0 or 1 density for some j . However, in this unstable case, since the corresponding column \vec{a}_{re} of A trends to 0, the redundant component $y^{(re)}$ has no effect to the reconstruction data x and cannot converge. Thus, the corresponding q_{re} cannot converge either and cannot be discarded.

2.3 Combination of B- and BI-architecture for BFA with automatic model selection

From the analysis in sect.2.2.2 we find that this unstable case can be easily solved by B-architecture. Moreover, Table. 2.5 shows that the local minima in B-architecture can also be much avoided in BI-architecture. Therefore, we propose a method of combining B-architecture and BI-architecture to get better performance for automatic model selection learning for implementing BFA.

2.3.1 Combine B-architecture and BI-architecture

From sect.2.2.2, it can be found that there is a complementation between B-architecture and BI-architecture. That is, the local minima can be considerably reduced in BI-architecture, while the unstable case that often occurs in BI-architecture can converge.

Thus, we propose a new strategy to combine these two architectures for implementing one learning process for BFA with automatic model selection, described as following steps:

- step 1: Implementing learning process using BI-architecture in Eq. 2.8,
- step 2: Using the obtained parameter values in step 1 as the initial values for implementing B-architecture algorithm in Eq. 2.9. (2.10)

Just as the comparison in sect. 2.1.3, the running time of BI-architecture is much less than that of B-architecture. Table. 2.7 confirms this conclusion. Based on the result of BI-architecture, a subsequent learning using

B-architecture only needs a little time to converge. Conclusively, this combination method can save more running time than B-architecture. Sect.2.4.5 will show an experimental comparison of the three strategies, B-architecture, BI-architecture as well as the combination of them.

There is no free lunch. Although this combination method has greatly improved the correct rate of automatic model selection learning and usually can work well, it is not omnipotent for any case, which will be discussed in the next section.

2.3.2 Limitations of BI-architecture

In this section, some limitations of BI-architecture are discussed which usually occur when the variance of error and the mean value of data become too large. In these cases, BI-architecture and the combination strategy cannot work well any more and only B-architecture is suitable for BFA.

One limitation is about the true variance of error σ_0^2 . The learning result of BI-architecture is greatly affected by the variance of error. When the variance of error increases to more than 2, BI-architecture even cannot work anymore. However, B-architecture is not so vulnerable by σ_0^2 . Experiment 7 in the next section shows the comparison. Also, it is found that the results of BI-architecture can be affected by the value of c_0 . When it is too large, the BI-architecture cannot work. However, B-architecture is rarely affected by c_0 . Experiment in sect.2.4.6 shows the comparison of them.

2.4 Experiments

In this section extensive experiments are done to see if our heuristic methods for B-architecture and the combination strategy work well.

The observation data $x_t, t = 1, \dots, n$ are generated from $x_t = Ay_t + c + e_t$ with y_t randomly generated from a Bernoulli distribution and e_t randomly generated from $G(x|0, \sigma^2 I)$. Each element of A is generated from $G(x|0, 1)$. Experiments are repeated over 100 times to facilitate our observing on statistical behaviors and all the experimental results are derived from testing set.

2.4.1 Frequency of local minima occurring in B-architecture

This experiment shows frequency of the local minima in B-architecture we discussed in sect.2.2.1. In this experiment, the dimension of x is $d = 15$ and the dimension of y is $k = 5$. The noise variance σ^2 is equal to 0.5 and the sample size $n = 100$. Table. 2.3 shows the frequency that this local minima occurs in a B-architecture based on 100 experiments. You can find that this special local minima occurs much more often than the the correct case, as well as other local cases.

Table 2.3: Rates of the special local minima (SL), other local (OL) and correct cases (C) in B-architecture in 100 experiments

C	SL	OL	Dimensionality of x	Dimensionality of y
38%	45%	17%	15	5

2.4.2 Performance comparison for several methods in B-architecture

This experiment shows the performances of two heuristic methods and the improved algorithm proposed in sect.2.2.1. A same data set is used to the different methods. An proper initial learning rate of A is usually in the range of 0.1 to 0.6 experimentally. The experiment condition is the same as experiment in sect.2.4.1. Table. 2.4 is a comparison of the frequency that the local minima occur. After using the two heuristic methods, the frequency of the special local minima and other local minima rapidly decreases and the correct rate is improved from 38% to 55%.

Table 2.4: Rates of the special local minima (SL), other local (OL) and correct cases(C) in comparison of different methods by 100 experiments

Methodology	C	SL	OL	Initial step length of A
original B-architecture	38%	45%	17%	0.01
B-architecture with two heuristic methods	55%	32%	13%	0.5
Improved B-architecture with two heuristic methods	87%	0	13%	0.5

2.4.3 Comparison of local minima in B-architecture and BI-architecture

In this experiment, you can easily find that local minima in BI-architecture are much less than that in B-architecture. The data is still 15-dimension and the true binary factors number k is 5. The same data is used to both B- and BI-architecture. Table. 2.5 shows an comparison of the frequency that local minima occur in B- and BI- architecture. In this table, you can also find that

the run time of BI-architecture is much less than that of B-architecture. It should be noted that the time given in Table 2.5 is an average of 100 experiments.

Table 2.5: Rates of the local minima in original B-architecture and BI-architecture in 100 experiments

Architecture	local minima	CPU time (in minutes)
original B-architecture	62%	5.2
BI-architecture	9%	1.3

2.4.4 Frequency of unstable cases occurring in BI-architecture

This experiment shows the frequency that the unstable cases occur in a BI-architecture. We use the same data as that of the experiment in sect.2.4.1. In table. 2.6, you can find that such unstable case described in sect.2.2.2 has high frequency 33%.

Table 2.6: Rates of unstable cases (US), local minima (L) and correct cases (C) in BI-architecture by 100 experiments

C	US	L	Dimensionality of x	Dimensionality of y
60%	30%	9%	15	5

2.4.5 Comparison of performance of three strategies

This experiment shows an performance comparison of three strategies, improved B-architecture in Eq. 2.9, BI-architecture in Eq. 2.8, as well as the combination of them in Eq. 2.10. A same data set is used to all these three strategies. The data set is the same as that in sect.2.4.1. Table. 2.7 shows

that the combination method has the highest correct rate with reasonable running time. Moreover, local minima has been considerably reduced by the combination method.

Table 2.7: Performance comparison of the three strategies, B-architecture, BI-architecture, and the combination strategy

Strategy	Correct result	Local minima	CPU time (in minutes)
B-architecture	87%	13%	5.2
BI-architecture	60%	6%	1.3
Combination	94%	6%	2.5

2.4.6 Limitations of BI-architecture

This experiment shows the limitations of BI-architecture mentioned in sect.2.3.2. One is about the variance of error represented by σ_0^2 , and the other is about the mean value of data represented by c_0 .

Table. 2.8 shows the reconstruction errors of x using B-architecture and BI-architecture with the variance of error σ_0^2 increasing from 0.1 to 1.5. The data used in this experiment is 15-dimension vector and there are 5 real binary factors. The data size is $n = 100$. You can easily find the difference between these two architectures in this table. The result using BI-architecture is seriously effected by σ_0^2 . In this table, when σ_0^2 increases to 1.5, the BI-architecture model even cannot work any more. However, the B-architecture is not so seriously affected by σ_0^2 .

Also, it is found that the results of BI-architecture are closely related to the mean value c_0 . When the mean value of data is too large, the BI-architecture

Table 2.8: Average results as σ_0^2 increases

B-architecture			
Para.	$\sigma_0 = 0.1$	$\sigma_0 = 1$	$\sigma_0 = 1.5$
Average error of x	0.0858	0.2164	0.6473
Iteration number	200	500	1000
BI-architecture			
Para.	$\sigma_0 = 0.1$	$\sigma_0 = 1$	$\sigma_0 = 1.5$
Average error of x	0.0869	0.6192	1.890
Iteration number	2000	6000	>20000

cannot even work. However, B-architecture is rarely affected by c_0 . This is presented in table. 2.9.

Table 2.9: Average results as the mean value of x increases.

B-architecture			
Para.	$Average_{c_0} = 0.58$	$Average_{c_0} = 5.46$	$Average_{c_0} = 18.3$
Average error of x	0.0642	0.0693	0.0711
Iteration number	300	300	300
BI-architecture			
Para.	$\sigma_0 = 0.1$	$\sigma_0 = 1$	$\sigma_0 = 1.5$
Average error of x	0.0802	0.7533	4.5361
Iteration number	2000	6000	Inf

2.5 Summary

In this chapter, we concentrated on the ability of automatic model selection for the BYY learning for implementing BFA. We mainly analyzed the local

minima problems in B-architecture and BI-architecture.

For B-architecture, the local minima that often happens has been described. Furthermore, two heuristic methods to avoid such local minima were proposed and an improved algorithm to remove this local minima was given. Experiments have proved that these methods are effective.

For BI-architecture, we also summarized the special cases often appearing in experiments. One unstable case was emphasized because it often occurs and can be rescued by B-architecture. Thus, we proposed a combination strategy for making use of the advantages of both adequately. Experiments have shown that this combination strategy efficiently avoids the local minima in B-architecture and solves the unstable problem in BI-architecture. Moreover, the running time of the combination method is much less than that of B-architecture.

Chapter 3

A Comparative Investigation on Model Selection in Binary Factor Analysis

This chapter investigates the BYY criterion and BYY harmony learning with automatic model selection (BYY-AUTO) in comparison with existing typical criteria, including AIC, BIC, CAIC, and CV criterion. This study is made via experiments on data sets with different sample sizes, data space dimensions, noise variances, and hidden factors numbers. Experiments have shown that the performance of BIC is superior to AIC, CAIC, and CV in most cases. In case of high dimensional data or large noise variance, the performance of CV is superior to AIC, CAIC, and BIC. The BYY criterion and the BYY automatic model selection learning (BYY-AUTO) are, in most cases, comparable with or even superior to the best among of BIC, AIC, CAIC, and CV. Moreover, selection of hidden factors number k by BIC, AIC, CAIC, and CV has to be made at the second stage on a set of candidate factor models obtained via

parameter learning at the first stage. This two-phase procedure is very time-consuming, while BYY-AUTO takes much less time than the conventional two-phase methods because an appropriate factors number k can be automatically determined during parameter learning.

3.1 Binary Factor Analysis and ML Learning

In sect. 2.1.1, we have introduced the basic issues of binary factor analysis (BFA). The task of model selection for BFA is to determine the hidden factors number k .

Given k and a set of observations $\{x_t\}_{t=1}^n$, one widely used method for estimating $\theta = \{A, c, \sigma^2\}$ is the maximum likelihood learning. That is,

$$\hat{\theta} = \arg \max_{\theta} L(\theta). \quad (3.1)$$

where $L(\theta)$ is the following log likelihood function

$$\begin{aligned} L(\theta) &= \sum_{t=1}^n \ln(p(x_t)) \\ &= \sum_{t=1}^n \ln\left(\sum_{y \in D} p(x_t|y)p(y)\right), \end{aligned} \quad (3.2)$$

where D is the set that contains all possible values of y .

This optimization problem can be implemented by EM algorithm that iterates following steps [7, 34]:

step 1: calculate $p(y|x_t)$ by

$$p(y|x_t) = \frac{p(x_t|y)p(y)}{\sum_{y \in D} p(x_t|y)p(y)}. \quad (3.3)$$

step 2: update A , c and σ^2 by

$$A = \left(\sum_{t=1}^n \sum_{y \in D} p(y|x_t)(x_t - c)y^T \right) \left(\sum_{t=1}^n \sum_{y \in D} p(y|x_t)yy^T \right)^{-1}, \quad (3.4)$$

$$c = \frac{1}{n} \sum_{t=1}^n \sum_{y \in D} p(y|x_t)(x_t - Ay), \quad (3.5)$$

and

$$\sigma^2 = \frac{1}{dn} \sum_{t=1}^n \sum_{y \in D} p(y|x_t) \|e_t\|^2 \quad (3.6)$$

respectively, where $e_t = x_t - Ay - c$.

3.2 Hidden Factors Number Determination

3.2.1 Using Typical Model Selection Criteria

As described in sect. 1.1.3, determination of hidden factors number k for BFA can be performed via several existing statistical model selection criteria. Such criteria include AIC, BIC, CAIC, CV, as well as MDL which formally coincides with BIC. These criteria have to be implemented in a two-phase procedure that is actually very time-consuming. First, we need to assume a range of values of k from k_{min} to k_{max} which is assumed to contain the optimal k . At each specific k , we estimate the parameters θ under the ML learning principle. Second, we make the following selection

$$\hat{k} = \arg \min_k \{J(\hat{\theta}, k), k = k_{min}, \dots, k_{max}\}, \quad (3.7)$$

where $J(\hat{\theta}, k)$ is a given model selection criterion.

Eq. 1.3 in sect.1.1.3 gives the general form of the three model selection criteria AIC, CAIC, and BIC. Applying these criteria to BFA, $L(\hat{\theta})$ is the log

likelihood Eq. 3.2 based on the ML estimation $\hat{\theta}$ using the EM algorithm in Eq. 3.3 - Eq. 3.6 under a given k , and $Q(k) = (d-1)k + d + 2$ is the number of free parameters in k -factors model of BFA. $B(n)$ has been given in sect.1.1.3.

Applying another well-known model selection technique, cross-validation (CV) in Eq. 1.4, to the model selection of BFA, the cross-validated log-likelihood for a k -factors model is

$$J(\hat{\theta}, k) = -\frac{1}{m} \sum_{i=1}^m L(\hat{\theta}(U_{-i})|U_i) \quad (3.8)$$

3.2.2 Using BYY harmony Learning

Determining hidden factors by BYY criterion

Applying BYY harmony learning to the binary factor analysis model, the following criterion is obtained for selecting the hidden factors number k [39]

$$J(\hat{\theta}, k) = k \ln 2 + 0.5d \ln \hat{\sigma}^2. \quad (3.9)$$

Shortly, we refer it by BYY criterion for BFA, where $\hat{\sigma}^2$ can be obtained via BYY harmony learning, i.e., Eq. 2.3 in sect.2.1.2 which is implemented by either a batch or an adaptive algorithm.

In sect.2.1.3, the adaptive algorithm of BYY harmony learning for BFA with automatic model selection has been given in Eq. 2.6. Now we give the parameter learning algorithm of BYY harmony learning for implementing BFA with two-phrase style. In a two-stage implementation, the parameters need to be learned include A , c , and σ^2 , the probability q_j for each hidden factor $y^{(j)}$ is prefixed as 0.5.

With k fixed, Eq. 2.4 can be implemented via the adaptive algorithm given by Table.1 in [39]. Considering that typical model selection criteria are evaluated basing on the ML estimation via the EM algorithm made in batch, we also implement Eq. 2.4 in batch. Similar to the procedure given in Table.1 of [39], we iterate the following steps:

$$\begin{aligned}
 \text{step 1:} & \quad \text{get } y_t \text{ by Eq. 2.5,} \\
 \text{step 2:} & \quad \text{from } \frac{\partial H(\theta, k)}{\partial \theta} = 0, (\theta \text{ include } A, c, \sigma^2), \text{ update} \\
 & \quad e_t = x_t - Ay_t - c, \\
 & \quad \sigma^2 = \frac{\sum_{t=1}^n \|e_t\|^2}{dn}, \\
 & \quad A = \left(\sum_{t=1}^n (x_t - c)y_t^T \right) \left(\sum_{t=1}^n y_t y_t^T \right)^{-1}, \\
 & \quad c = \frac{1}{n} \sum_{t=1}^n (x_t - Ay_t), \tag{3.10}
 \end{aligned}$$

This iterative procedure is guaranteed to converge since it is actually the specific form of the Ying-Yang alternative procedure, see Sect.3.2 in [38].

With k enumerated and its corresponding parameters obtained by the above Eq. 3.10, we can select a best value of k by BYY criterion for BFA in Eq. 3.9.

Automatic hidden factor determination

Furthermore, an adaptive algorithm has also been developed from implementing BYY harmony learning with appropriate hidden factors automatically determined during learning [36, 38, 39]. Instead of prefixing q_j , q_j is learned together with other parameters θ via maximizing $H(\theta, k)$, which will lead to that q_j on each extra dimension $y^{(j)}$ is pushed towards 0 or 1 and thus we

can discard the corresponding dimension [36, 38, 39]. Setting k initially large enough (e.g., the dimensionality of x in this paper), we can implement the adaptive algorithm with model selection made automatically. Shortly, we refer it by BYY automatic model selection learning (BYY-AUTO). The adaptive algorithm of BYY harmony learning with automatic model selection has been given in sect.2.1.3 and sect.2.1.4 for two architectures respectively. In experiments, we used a combination strategy Eq.2.10 via BYY automatic model selection learning (BYY-AUTO) in chapter 2 for getting higher correction rate and better performance for the implementation of BFA.

3.3 Empirical Comparative Studies

We investigate the experimental performances of the model selection criteria AIC, BIC, CAIC, 10-fold CV, BYY criterion and BYY-AUTO on four types of data sets with different sample sizes, data dimensions, noise variances, and numbers of hidden factors. In implementation, for AIC, BIC, CAIC and 10-fold CV, we use EM algorithm Eq. 3.3- Eq. 3.6 to obtain the ML estimates of A , c and σ^2 . For BYY criterion we implement algorithm Eq. 3.10 for parameter learning. For BYY-AUTO, we use the combination algorithm Eq. 2.10 for the most cases except the case when noise variance σ^2 becomes large to 1.5 and for this special case we use the algorithm Eq. 2.6 of B-architecture for automatic model selection during parameter learning. The observations x_t , $t = 1, \dots, n$ are generated from $x_t = Ay_t + c + e_t$ with y_t randomly generated from a Bernoulli distribution with q_j is equal to 0.5 and e_t randomly generated from $G(x|0, \sigma^2 I)$. Experiments are repeated over 100 times to facilitate

our observing on statistical behaviors. Each element of A is generated from $G(x|0, 1)$. Usually we set $k_{min} = 1$ and $k_{max} = 2k - 1$ where k is the true number of hidden factors. In addition, to clearly illustrate the curve of each criterion within a same figure we normalize the values of each curve to zero mean and unit variance.

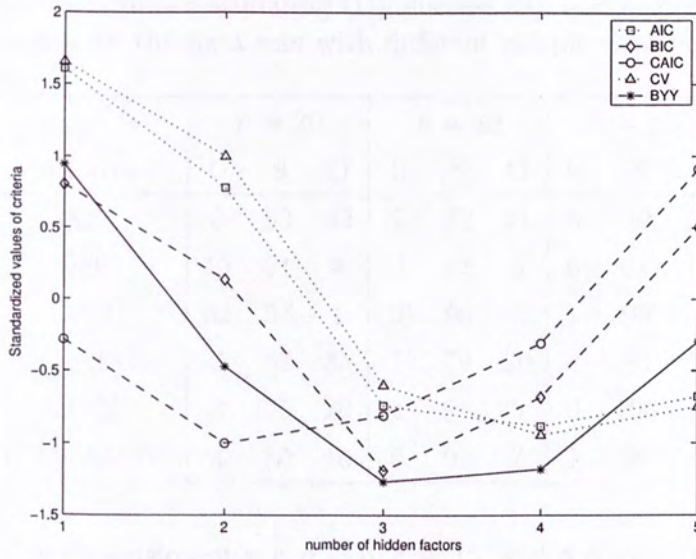
3.3.1 Effects of Sample Size

We investigate the performances of every method on the data sets with different sample sizes $n = 20$, $n = 40$, and $n = 100$. In this experiment, the dimension of x is $d = 9$ and the dimension of y is $k = 3$. The noise variance σ^2 is equal to 0.1. The results are shown in Fig. 3.1. Table 3.1 illustrates the rates of underestimating, success, and overestimating of each method in 100 experiments.

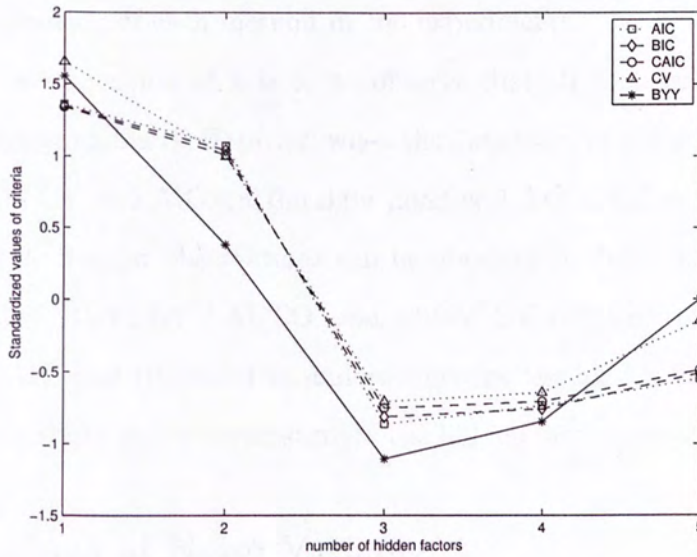
When the sample size is only 20, we see that BYY and BIC select the right number 3. CAIC selects the number 2. AIC, 10-fold CV select 4. When the sample size is 100, all the criteria lead to the right number. Similar observations can be observed in Table 3.1. For a small sample size, CAIC tends to underestimate the number while AIC, 10-fold CV tend to overestimate the number. BYY criterion has a little risk of overestimation, and BYY-AUTO is comparable with BIC.

3.3.2 Effects of Data Dimension

Next we investigate the effect of data dimension on each method. The dimension of y is $k = 3$, the noise variance σ^2 is equal to 0.1, and the sample size



(a) $n = 20$



(b) $n = 100$

Figure 3.1: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY on the data sets of a 9-dimensional x ($d = 9$) generated from a 3-dimensional y ($k = 3$) with different sample sizes for BFA.

Table 3.1: Rates of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different sample sizes for BFA in 100 experiments

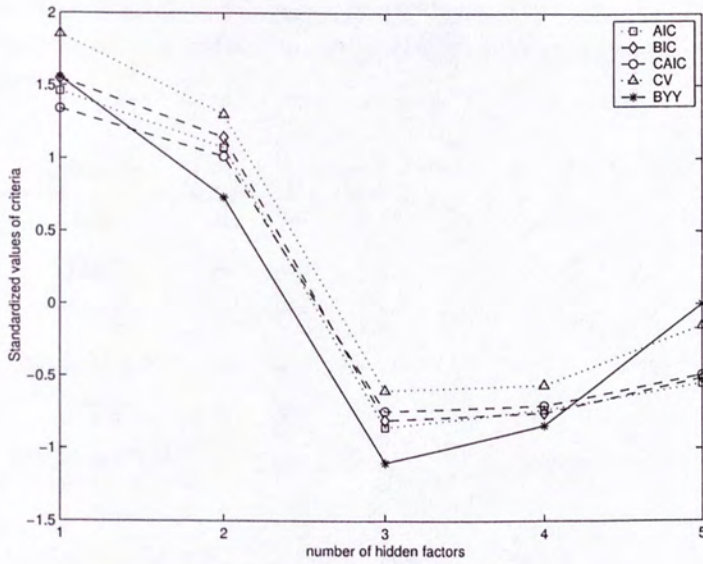
criteria	$n = 20$			$n = 40$			$n = 100$		
	U	S	O	U	S	O	U	S	O
AIC	5	53	42	1	78	21	0	89	11
BIC	15	81	4	3	97	0	0	100	0
CAIC	32	67	1	10	90	0	1	99	0
10-fold CV	3	62	35	1	79	20	0	90	10
BYY	3	75	20	1	94	5	0	100	0
BYY-AUTO	4	80	16	0	93	7	1	99	0

is $n = 80$. The dimension of x is $d = 6$, $d = 15$, and $d = 25$. The results are shown in Fig. 3.2. Table 3.2 illustrates the rates of underestimating, success, and overestimating of each method in 100 experiments.

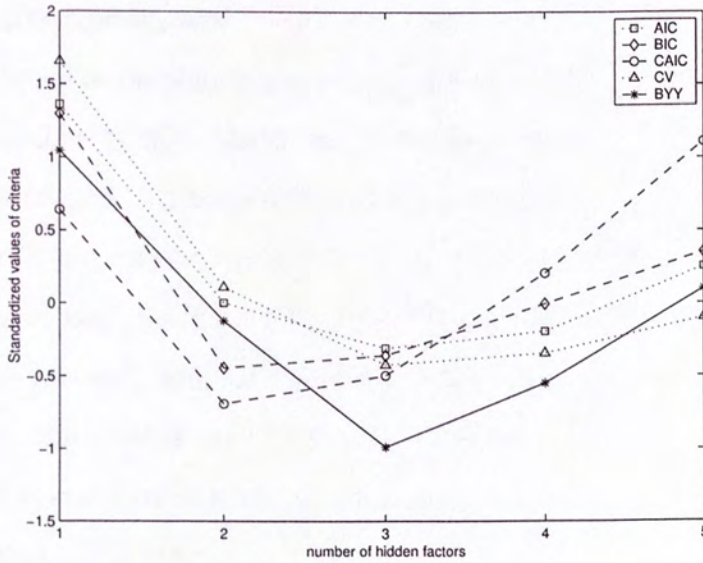
When the dimension of x is 6, we observe that all these criteria tend to select the right number 3. However, when the dimension of x is increased to 25, BYY, 10-fold CV and AIC get the right number 3, but CAIC and BIC choose the number 2. Similar observations can be obtained in Table 3.2. For a high dimensional x , BYY, BYY-AUTO, and 10-fold CV still have high successful rates but CAIC and BIC tend to underestimating the hidden factors number k . AIC has a slight risk to overestimate the hidden factors number.

3.3.3 Effects of Noise Variance

We further investigate the performance of each method on the data sets with different scales of noise added. In this example, the dimension of x is $d = 9$, the dimension of y is $k = 3$, and the sample size is $n = 80$. The noise variance



(a) $d = 6$



(b) $d = 25$

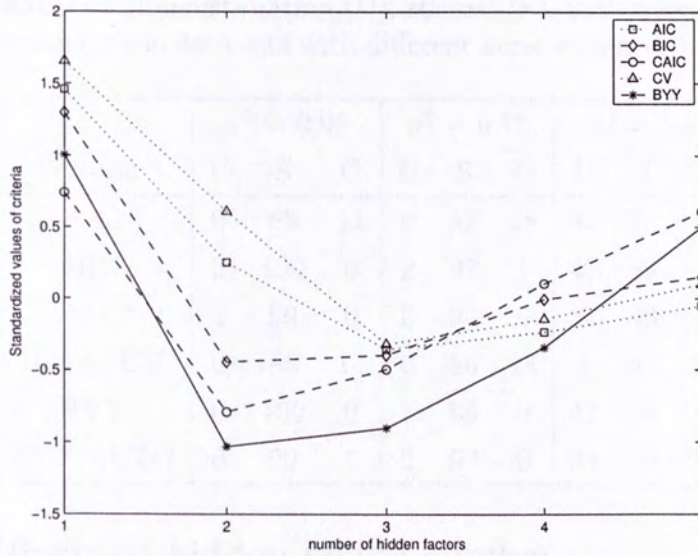
Figure 3.2: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY on the data sets of a x with different dimensions generated from a 3-dimensional y ($k = 3$) for BFA.

Table 3.2: Rates of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different data dimensions for BFA in 100 experiments

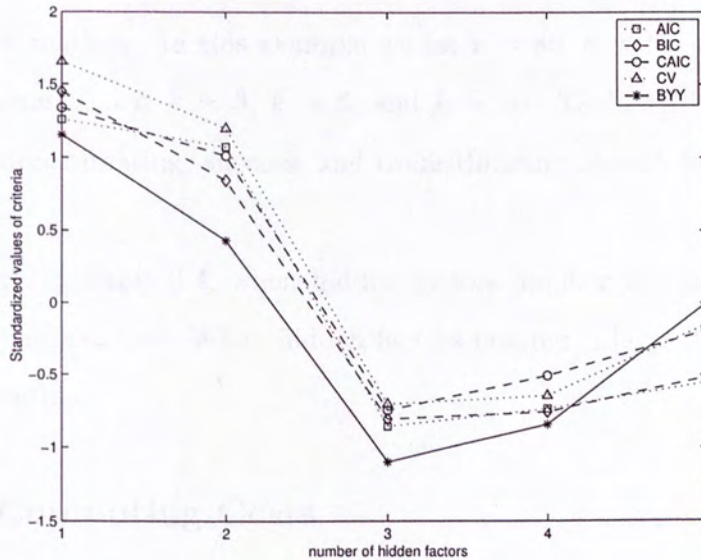
criteria	$d = 6$			$d = 15$			$d = 25$		
	U	S	O	U	S	O	U	S	O
AIC	0	89	11	0	86	14	2	83	16
BIC	0	98	2	3	96	1	30	69	1
CAIC	0	100	0	7	93	0	48	52	0
10-fold CV	0	90	10	0	86	14	0	89	11
BYY	0	99	1	1	95	4	10	89	1
BYY-AUTO	1	99	0	1	93	6	3	87	10

σ^2 is equal to 0.05, 0.75, and 1.5. The results are shown in Fig. 3.3. Table 3.3 illustrates the rates of underestimating, success, and overestimating of each method in 100 experiments.

When the noise variance is 1.5, we see that only AIC and 10-fold CV select the right number 3, BIC, CAIC and BYY select 2 factors. When the noise variance is 0.05 or 0.75, all the criteria lead to the right number. Similar observations can be observed in Table 3.3. From this table we can find, for a large noise variance, BIC, CAIC, BYY, and BYY-AUTO are high likely to underestimate the number, AIC and 10-fold CV have a slight risk of overestimate the number. For a small noise variance, CAIC, BIC, BYY and BYY-AUTO have high successful rates while AIC and 10-fold CV still have a slight risk of overestimating the number.



(a) $\sigma^2 = 1.5$



(b) $\sigma^2 = 0.05$

Figure 3.3: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY on the data sets of a 9-dimensional x ($d = 9$) generated from a 3-dimensional y ($k = 3$) with different noise variances for BFA.

Table 3.3: Rates of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different noise variances for BFA in 100 experiments

criteria	$\sigma^2 = 0.05$			$\sigma^2 = 0.75$			$\sigma^2 = 1.5$		
	U	S	O	U	S	O	U	S	O
AIC	0	89	11	0	82	18	6	78	16
BIC	0	100	0	2	97	1	40	58	2
CAIC	1	99	0	6	94	0	57	43	0
10-fold CV	0	86	14	0	86	14	1	81	18
BYE	0	100	0	1	99	0	42	52	6
BYE-AUTO	0	99	1	2	98	0	24	50	26

3.3.4 Effects of hidden factor number

Finally, we consider the effect of hidden factor number, that is, the dimension of y on each method. In this example we set $n = 80$, $d = 15$, and $\sigma^2 = 0.1$. The dimension of y is $k = 3$, $k = 6$, and $k = 10$. Table 3.4 illustrates the rates of underestimating, success, and overestimating of each method in 100 experiments.

As shown in Table 3.4, when hidden factors number is small all criteria have good performance. When hidden factors number is large AIC get a risk of overestimating.

3.3.5 Computing Costs

All the experiments were carried out using MATLAB R12.1 v.6.1 on a P4 2GHz 512KB RAM PC. We list the computational results in Table 3.5 for the first example described in sect. 2.4.1 with a sample size $n = 100$. For AIC,

Table 3.4: Rates of underestimating (U), success (S), and overestimating (O) by each criterion on simulation data sets with different hidden factors numbers for BFA in 100 experiments

criteria	$k = 3$			$k = 6$			$k = 10$		
	U	S	O	U	S	O	U	S	O
AIC	0	86	14	0	85	15	0	72	28
BIC	2	98	0	4	96	0	9	91	0
CAIC	6	94	0	9	91	0	11	89	0
10-fold CV	0	85	15	0	85	15	0	81	19
BYY	2	98	0	4	95	1	10	86	4
BYY-AUTO	1	99	0	4	93	0	11	86	3

BIC, CAIC and 10-fold CV, we list the total time of implementing the EM algorithm and of evaluating the criteria by $k_{max} - k_{min} + 1 = 5$ times. For BYY criterion we list the total time of implementing the algorithm Eq. 3.10 and of evaluating the criteria also by $k_{max} - k_{min} + 1 = 5$ times. For BYY-AUTO, we list the time of implementing the algorithm Eq. 2.6 as well as Eq. 2.8 to make initialization. It should be noted that all the values given in Table 3.5 are the average of 100 experiments.

Moreover, in Table 3.6 we list the running time of both BYY harmony learning by Eq. 3.10 and EM algorithm Eq. 3.3- Eq. 3.6 when $k = 3$ using the same data as above. This table shows that the the algorithm derived from BYY harmony learning takes much less time than the conventional EM algorithm.

The hidden factor number determination by AIC, BIC, CAIC takes a similar CPU time and takes more time than that by BYY criterion since the EM algorithm needs more iterations than the algorithm Eq. 3.10 needs. The

Table 3.5: CPU times on the simulation data sets with $n = 100$, $d = 9$, and $k = 3$ for BFABY using the EM algorithm for AIC, BIC, CAIC and CV, algorithm (3.10) for BYY criterion and (2.6) for BYY-AUTO

method	CPU time (in minutes)
BYY-AUTO	1
BYY criterion	5.2
AIC, CAIC, BIC	15.3
10-fold CV	58.5

Table 3.6: CPU times on the simulation data sets with $n = 100$, $d = 9$, and $k = 3$ for BFA by using the EM algorithm and the algorithm derived from BYY harmony learning where set the candidate $k = 3$

method	CPU time (in minutes)
BYY harmony learning	0.9
EM algorithm	2.2

m -fold cross-validation method consumed the highest computing cost because parameters have to be estimated by m times on each candidate model. The computing costs of all these criteria are much higher than that of BYY-AUTO. In a summary, the performances by BYY criterion and BYY-AUTO are either superior or comparable to those by typical statistical model selection criteria, while BYY-AUTO saves computing costs considerably. That is, BYY harmony learning is much more favorable.

3.4 Summary

We have made experimental comparisons on determining the hidden factor number in implementing BFA. The methods include four typical model selection criteria AIC, BIC, CAIC, 10-fold CV, and BYY criterion as well as BYY learning with automatic model selection. We have observed that the performances by BYY criterion and BYY-AUTO are either superior or comparable to other methods in most cases. Both BYY criterion and BYY-AUTO have high successful rates except the cases of large noise variance. BIC also got a high successful rate when the data dimension is not too high and the noise variance is not too large. CAIC has an underestimation tendency while AIC and 10-fold CV have an overestimation tendency. Moreover, BYY-AUTO needs a much less computing time than all the considered criteria including BYY criterion.

Chapter 4

A Comparative Investigation on Model Selection in Non-gaussian Factor Analysis

Non-Gaussian factor analysis (NFA) is a recently proposed technique for the multivariate data analysis with non-Gaussian latent factors. A crucial issue in NFA is the determination of the hidden factors number and the model complexity of each factor. This chapter investigates the BYY criterion in comparison with existing typical criteria, AIC, BIC, CAIC, and CV, on the model selection of NFA. This comparative study is made via experiments on the data sets with different sample sizes, data space dimensions, and noise variances. Experiments have shown that in most cases BIC outperforms AIC, CAIC, and CV while the BYY criterion is either comparable with or better than BIC. Furthermore, the algorithm derived from BYY harmony learning takes much less time than the conventional EM algorithm since the computational complexity grows exponentially with the number of factors in EM algorithm.

4.1 Non-Gaussian Factor Analysis and ML Learning

Non-Gaussian factor analysis (NFA) generalizes the classic factor analysis (FA) Eq. 1.1 by assuming y is a non-Gaussian random vector [4, 40]. In this paper we consider each factor $y^{(j)}$ is derived from a Gaussian mixture distribution [20, 36, 38]

$$p(y) = \prod_{j=1}^k [p_j(y^{(j)})], \quad p_j(y^{(j)}) = \sum_{q_j=1}^{k_j} \beta_{j,q_j} G(y^{(j)} | \mu_{j,q_j}, \sigma_{j,q_j}^2) \quad (4.1)$$

where $G(y^{(j)} | \mu_{j,q_j}, \sigma_{j,q_j}^2)$ denotes a normal (Gaussian) distribution with mean μ_{j,q_j} and variance σ_{j,q_j}^2 . We consider e is drawn from a Gaussian distribution, thus $p(x|y)$ has the following form:

$$p(x|y) = G(x|Ay, \Sigma), \quad (4.2)$$

where Σ is the covariance matrix of error e .

For simplification, we preprocess the observable data to be zero mean such that the unknown parameter c in Eq. 1.1 can be ignored. In implementation of NFA, the unknown parameters θ consists of the mixing matrix A , the covariance matrix Σ , and the parameters $\theta_j = \{\beta_{j,q_j}, \mu_{j,q_j}, \sigma_{j,q_j}^2\}$ for each factor $y^{(j)}$. Given a number k of factors and the number k_j for each factor $y^{(j)}$ (denoted by $K = \{k, \{k_j\}\}$) as well as a set of observations $\{x_t\}_{t=1}^n$, maximum likelihood learning by Eq. 3.1 is still used for estimating θ , with the following log likelihood function:

$$L(\theta) = \sum_{t=1}^n \ln(p(x_t))$$

$$= \sum_{t=1}^n \ln \left(\int p(x_t|y)p(y)dy \right), \quad (4.3)$$

which can not be implemented by the EM algorithm as the integral in this function is analytically intractable. In [20], a missing data $q = [q_1, q_2, \dots, q_k]$ is used to indicate which factor is generated by the corresponding Gaussian component, and then $p(y)$ in Eq. 4.5 is expressed as a mixture of Gaussian products. As a result, the integral becomes a summation of a large number of analytically computable integrals on Gaussians, which makes an exact ML learning on Eq. 1.1 implemented by an exact EM algorithm. The same approach has been also published in [4] under the name of independent factor analysis.

Specifically, each state q_j in q indicates the factor $y^{(j)}$ generated by the q_j th Gaussian component and each state q corresponds to a k -dimensional Gaussian density with the mixing proportions β_q , mean μ_q , and diagonal covariance matrix V_q as follows:

$$\begin{aligned} \beta_q &= \prod_{j=1}^k \beta_{j,q_j} = \beta_{1,q_1} \cdot \dots \cdot \beta_{k,q_k}, \\ \mu_q &= [\mu_{1,q_1}, \dots, \mu_{k,q_k}]^T, \\ V_q &= \text{diag}(\sigma_{1,q_1}^2, \dots, \sigma_{k,q_k}^2), \end{aligned} \quad (4.4)$$

where the notation $\text{diag}(d_1, \dots, d_k)$ denotes a diagonal matrix with the diagonal elements being d_1, \dots, d_k .

Thus, the form of $p(y)$ in Eq. 4.5 can be rewritten as

$$p(y) = \sum_q \beta_q G(y|\mu_q, V_q), \quad (4.5)$$

where the summation $\sum_q = \sum_{q_1}, \dots, \sum_{q_k}$. Also, the form of $p(q)$ and $p(x|q)$

can be written as

$$\begin{aligned} p(q) &= \beta_q, \\ p(x|q) &= G(x|A\mu_q, AV_qA^T + \Sigma). \end{aligned} \quad (4.6)$$

The EM algorithm for solving Eq. 4.3 is given in the following steps [20, 4]:

step E: calculate $p(q|x_t)$ by

$$p(q|x_t) = \frac{p(q)p(x_t|q)}{\sum_q p(q)p(x_t|q)}, \quad (4.7)$$

$$p(q_j|x_t) = \sum_{\{q_i\}_{i \neq j}} p(q|x_t), \quad (4.8)$$

where $\sum_{\{q_i\}_{i \neq j}}$ denotes summation over $\{q_{i \neq j}\}$, holding q_j fixed.

step M: update A , Σ and β_{j,q_j} , μ_{j,q_j} , σ_{j,q_j}^2 by

$$A = \sum_{t=1}^n x_t \langle y_t^T \rangle \left(\sum_{t=1}^n \langle y_t y_t^T \rangle \right)^{-1}, \quad (4.9)$$

$$\Sigma = \frac{1}{n} \sum_{t=1}^n x_t x_t^T - \frac{1}{n} \sum_{t=1}^n x_t \langle y_t^T \rangle A^T, \quad (4.10)$$

$$\mu_{j,q_j} = \frac{\sum_{t=1}^n p(q_j|x_t) \langle y_t^{(j)} \rangle}{\sum_{t=1}^n p(q_j|x_t)}, \quad (4.11)$$

$$\sigma_{j,q_j}^2 = \frac{\sum_{t=1}^n p(q_j|x_t) \langle y_t^{(j)2} \rangle}{dn} - \mu_{j,q_j}^2, \quad (4.12)$$

$$\beta_{j,q_j} = \frac{1}{n} \sum_{t=1}^n p(q_j|x_t), \quad (4.13)$$

respectively, where

$$\langle y_t^T \rangle = \sum_q p(q|x_t) h_q,$$

$$\begin{aligned}
 \langle y_t y_t^T \rangle &= \sum_q p(q|x_t)(h_q h_q^T + M_q), \\
 \langle y_t^{(j)} \rangle &= \sum_{\{q_i\}_{i \neq j}}^q p(q|x_t)(h_q)_j, \\
 \langle y_t^{(j)2} \rangle &= \sum_{\{q_i\}_{i \neq j}} p(q|x_t)(h_q h_q^T + M_q)_{jj}, \\
 M_q &= (A^T \Sigma^{-1} A + V_q^{-1})^{-1}, \\
 h_q &= M_q (A^T \Sigma^{-1} x_t + V_q^{-1} \mu_q),
 \end{aligned} \tag{4.14}$$

where $(h_q)_j$ means the j th element of vector h_q .

Obviously, the number of terms in the summation $\sum_q = \sum_{q_1}, \dots, \sum_{q_k}$ grows exponentially with k , and correspondingly incurs that computing costs exponentially grows with k . This is a serious disadvantage of this approach.

4.2 Hidden Factor Determination

4.2.1 Using typical model selection criteria

Determination of $K = \{k, k_j\}$ can be performed via several existing statistical model selection criteria such as AIC, BIC, CAIC, and CV which have to be implemented via a two-stage style. First, we need to assume a range of values of k from k_{min} to k_{max} and a range of values of each k_j from $k_{j_{min}}$ to $k_{j_{max}}$ which are assumed to contain the optimal k, k_j . This will construct a domain U for $K = \{k, k_j\}$. At each specific $K \in U$, we estimate the parameters θ under the ML learning principle. Second, we make the following selection

$$\hat{K} = \arg \min_K \{J(\hat{\theta}, K), K \in U\}, \tag{4.15}$$

where $J(\hat{\theta}, K)$ is a given model selection criterion.

Eq. 1.3 in sect.1.1.3 gives the general form of the three model selection criteria AIC, CAIC, and BIC. Applying these criteria to NFA, $L(\hat{\theta})$ is the log likelihood Eq. 4.3 based on the ML estimation $\hat{\theta}$ using the EM algorithm in Eq. 4.7 - Eq. 4.14 under a given K , and $Q(K) = (d - 2)k + 0.5d(d + 1) + \sum_{j=1}^k (3k_j - 1) + 1$ is the number of free parameters in K -model of NFA. $B(n)$ has been given in sect.1.1.3.

Applying another well-known model selection technique, cross-validation (CV) in Eq. 1.4, to the model selection of NFA, the cross-validated log-likelihood for a k -factors model is

$$J(\hat{\theta}, K) = -\frac{1}{m} \sum_{i=1}^m L(\hat{\theta}(U_{-i})|U_i) \quad (4.16)$$

In implementation, determining the hidden factors number k together with the Gaussians number k_j for each factor is a complex procedure since the domain U consists of all possible combinations of the values of k and each k_j . For simplification, we can determine them separately and set all factors Gaussians number k_j as a same integer. That is, hold the Gaussians number k_j fixed when determining the factors number k and fix k when determining the number k_j .

4.2.2 BYY harmony Learning

Applying BYY harmony learning to the non-Gaussian factor analysis model, the following criterion is obtained for selecting $K = \{k, k_j\}$ [38, 40]

$$J(\hat{\theta}, K) = \frac{1}{2} \ln |\hat{\Sigma}| + \frac{k}{2} (1 + \ln(2\pi)) + \sum_{j=1}^k \sum_{q_j=1}^{k_j} \hat{\beta}_{j,q_j} \left(\frac{1}{2} \ln \hat{\sigma}_{j,q_j}^2 - \ln \hat{\beta}_{j,q_j} \right). \quad (4.17)$$

Shortly, we refer it by BYY criterion for NFA where $\hat{\theta}$ can be obtained via BYY harmony learning by a batch or an adaptive algorithm, i.e.,

$$\hat{\theta} = \arg \max_{\theta} H(\theta, \hat{K}). \quad (4.18)$$

According to Sec. IV.A in [40], especially Eq. 52 in [40], the specific form of $H(\theta, K)$ is given as follows

$$\begin{aligned} H(\theta, K) &= -0.5 \ln |\Sigma| - \frac{1}{n} \sum_{t=1}^n \sum_{j=1}^k \ln p_j(y_t^{(j)}), \\ \Sigma &= \frac{1}{n} \sum_{t=1}^n e_t e_t^T, \\ e_t &= x_t - A y_t, \end{aligned} \quad (4.19)$$

where $p_j(y_t^{(j)})$ is given in Eq. 4.5 and $y_t^{(j)}$ is the j -th element in vector y_t , and

$$y_t = y(x_t) = \arg \max_y H(\theta, K). \quad (4.20)$$

Given x_t , y_t can be obtained via a nonlinear optimization algorithm. In this paper, we use the fixed posterior approximation to get y_t via iterating the following two steps [36, 42]:

$$\begin{aligned} \text{Step (a):} \quad p_{j,q_j} &= \frac{\beta_{j,q_j} G(y_t^{(j)} | \mu_{j,q_j}, \sigma_{j,q_j}^2)}{\sum_{q_j=1}^{k_j} \beta_{j,q_j} G(y_t^{(j)} | \mu_{j,q_j}, \sigma_{j,q_j}^2)}, \\ b_j &= \sum_{q_j=1}^{k_j} \frac{p_{j,q_j}}{\sigma_{j,q_j}^2}, d_j = \sum_{q_j=1}^{k_j} \frac{p_{j,q_j} \mu_{j,q_j}}{\sigma_{j,q_j}^2}, \\ \text{Step (b):} \quad y_t^{\text{new}} &= (A^T \Sigma^{-1} A + \text{diag}(b_1, \dots, b_k))^{-1} \\ &\quad \times (A^T \Sigma^{-1} x_t + [d_1, \dots, d_k]^T). \end{aligned} \quad (4.21)$$

With $K = \{k, k_j\}$ fixed, Eq. 4.19 can be implemented via the algorithm given by Eq. 57 in [40]. Similar to the procedure given in Eq. 57 of [40], we

iterate the following steps:

Yang step: get y_t by Eq. 4.21,

Ying step: (a) updating parameters in $p(x|y)$

$$e_t = x_t - Ay_t,$$

$$\Sigma^{\text{new}} = (1 - \eta)\Sigma^{\text{old}} + \eta e_t e_t^T,$$

$$A^{\text{new}} = A^{\text{old}} + \eta e_t y_t^T.$$

(b) updating parameters in $p(y)$

$$p_{j,q_j} = \frac{\beta_{j,q_j} G(y_t^{(j)} | \mu_{j,q_j}, \sigma_{j,q_j}^2)}{\sum_{q_j=1}^{k_j} \beta_{j,q_j} G(y_t^{(j)} | \mu_{j,q_j}, \sigma_{j,q_j}^2)},$$

$$\beta_{j,q_j}^{\text{new}} = (1 - \eta_0)\beta_{j,q_j}^{\text{old}} + \eta_0 p_{j,q_j},$$

$$\mu_{j,q_j}^{\text{new}} = \mu_{j,q_j}^{\text{old}} + \eta_0 p_{j,q_j} (y_t^{(j)} - \mu_{j,q_j}^{\text{old}}),$$

$$\sigma_{j,q_j}^{2\text{ new}} = (1 - \eta_0 p_{j,q_j})\sigma_{j,q_j}^{2\text{ old}} + \eta_0 p_{j,q_j} (y_t^{(j)} - \mu_{j,q_j}^{\text{old}})^2,$$

$$\mu_j = \sum_{q_j=1}^{k_j} \beta_{j,q_j}^{\text{new}} \mu_{j,q_j}^{\text{new}},$$

$$\sigma_j^2 = \sum_{q_j=1}^{k_j} \beta_{j,q_j}^{\text{new}} \sigma_{j,q_j}^{2\text{ new}},$$

$$\mu_{j,q_j}^{\text{new}} = \frac{(\mu_{j,q_j}^{\text{new}} - \mu_j)}{\sigma_j}, \sigma_{j,q_j}^{2\text{ new}} = \frac{\sigma_{j,q_j}^{2\text{ new}}}{\sigma_j^2}, \quad (4.22)$$

where η, η_0 are step length constants. In this paper, for simplification, we set

all k_j s as a same integer. This iterative procedure is guaranteed to converge since it is actually the specific form of the Ying-Yang alternative procedure, see Sec. III in [36].

With K enumerated as in Eq. 4.15 and its corresponding parameters obtained by the above Eq. 4.22, we can select a best K by BYY criterion in Eq. 4.17.

Besides the above criterion based selection, adaptive algorithm has also been developed from BYY harmony learning such that an appropriate K can be automatically determined during adaptively learning [36, 38, 40]. The hidden factors obtained via either this automatic determination or the above criterion have no difference. The difference is that the automatic determination saves significantly computational costs of implementing the conventional two stage style of statistical model selection. Thus, if the performances by the criterion from BYY harmony learning are comparable or even superior to typical statistical model selection criteria of AIC, CAIC, BIC, and CV in the most cases, we certainly prefer to use BYY harmony learning as a tool for determining hidden factors number k and gaussians number k_j .

4.3 Empirical Comparative Studies

We investigate the experimental performances of the model selection criteria AIC, BIC, CAIC, 10-fold CV and BYY criterion for NFA on three types of data sets with different sample sizes, data dimensions, and noise variances. In implementation, the EM algorithm Eq.4.7 - Eq. 4.14 is used to obtain ML estimates of all the parameters θ which consist of A , Σ and β_{j,q_j} , μ_{j,q_j} , σ_{j,q_j}^2 for

AIC, BIC, CAIC and 10-fold CV. For BYY criterion we implement algorithm Eq. 4.22 for parameters learning. The observations $x_t, t = 1, \dots, n$ are generated from $x_t = Ay_t + e_t$ with each $y_t^{(j)}$ randomly generated from a Gaussian mixture with 3 Gaussians and e_t randomly generated from $\mathcal{N}(0, \Sigma)$. Each element of A is generated from $\mathcal{N}(0, 1)$. Experiments are repeated over 50 times to facilitate our observing on statistical behaviors.

In our experiments, we assume all the numbers of Gaussian components k_j for factor $y^{(j)}$ are equal with each other thus K need to be determined via model selection criterion consists of two numbers k and k_j . For simplification, in our experiments k and k_j are determined separately. That is, hold the Gaussians number $k_j = 3$ fixed when determining the factors number k and the number k is given when determining the number k_j . Usually we set $k_{min} = 1$ and $k_{max} = 2k - 1$ where k is the true number of hidden factors and $k_{jmin} = 1$ and $k_{jmax} = 5$ since the true number of k_j is 3. In addition, to clearly illustrate the curve of each criterion within a same figure we normalize the values of each curve to zero mean and unit variance.

4.3.1 Effects of Sample Size on Model Selection Criteria

We investigate the performances of every criterion on the data sets with different sample sizes $n = 20, n = 40, \text{ and } n = 100$. In this experiment, the dimension of x is $d = 7$ and the dimension of y is $k = 3$. The noise covariance matrix Σ is equal to $0.1I$ (I is a 7×7 identity matrix). The results with different factors number k and fixing k_j as 3 are shown in Figure 4.1. The results

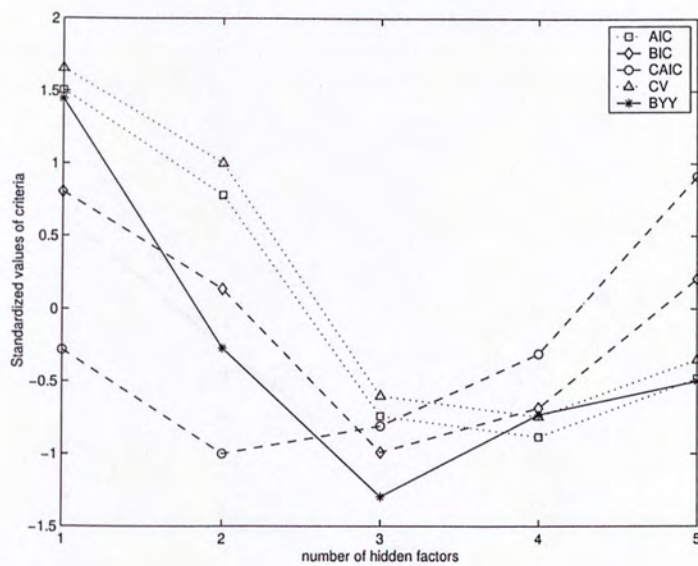
Table 4.1: Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different sample sizes for selecting hidden factors number k with $k_j = 3$ fixed for NFA in 50 experiments

criteria	$n = 20$			$n = 40$			$n = 100$		
	U	S	O	U	S	O	U	S	O
AIC	4	25	21	2	32	16	1	40	9
BIC	10	38	2	8	42	0	1	49	0
CAIC	19	31	0	14	36	0	4	46	0
10-fold CV	1	27	22	1	34	15	0	42	8
BYY	3	34	13	1	41	8	0	48	2

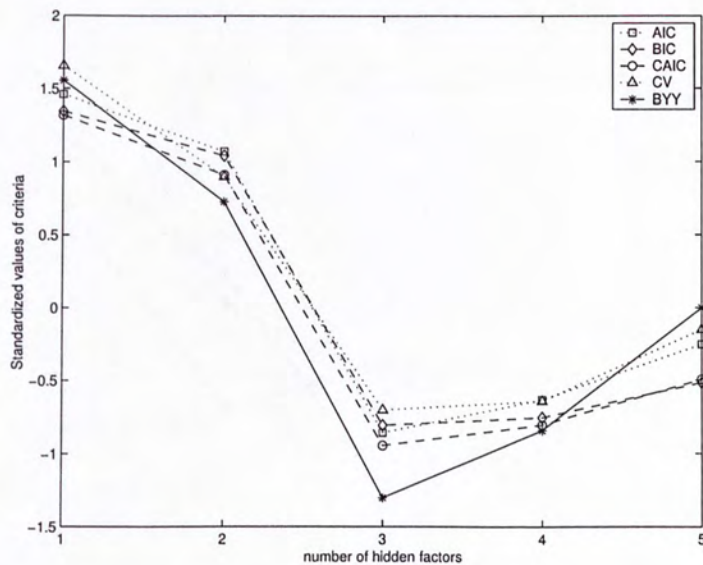
with different k_j and fixing k as 3 are shown in Figure 4.2. Table 4.1 and Table 4.2 illustrate the numbers of underestimating, success, and overestimating of each method for selecting k and k_j respectively in 50 experiments.

When the sample size is only 20, we see that BYY and BIC select the right hidden factors number 3. CAIC selects the number 2. AIC, 10-fold CV select 4. When the sample size is 100, all the criteria lead to the right number. Similar observations can be observed in Table 4.1. For a small sample size, CAIC tends to underestimate the number while AIC, 10-fold CV tend to overestimate the number. BYY criterion has a little risk of overestimation while BIC has a little risk of underestimation.

When the sample size is only 20, AIC and BYY select the right gaussians number 3 for each gaussian mixture of y_j . BIC select the number 2 and CAIC select the number 1. 10-fold CV select the number 4. When the sample size is 100, only CAIC leads to the number 2, all the other criteria select the right number 3. Similar observations can be observed in Table 4.2. CAIC tends to underestimate even the sample size is large enough.

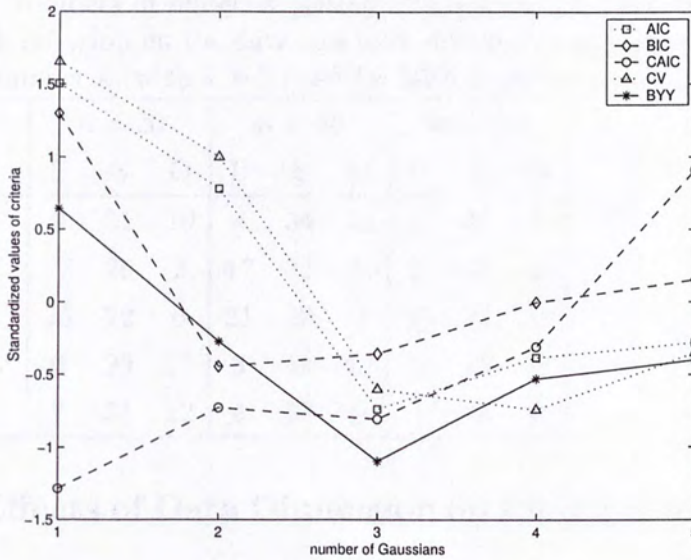


(a) $n = 20$

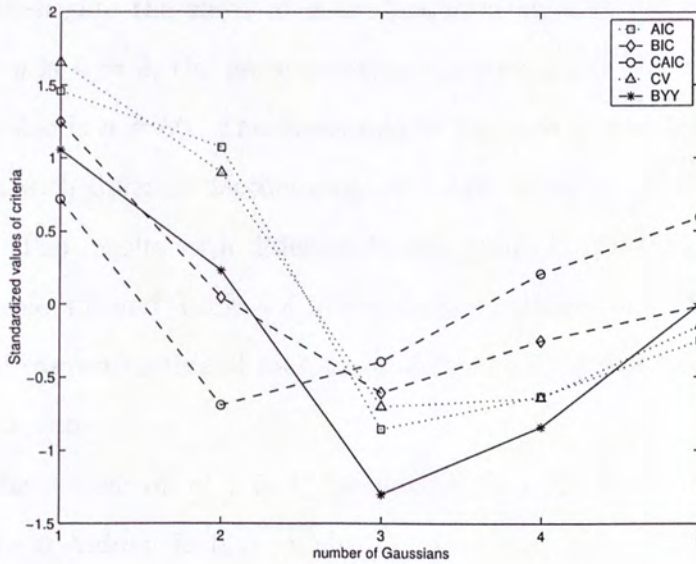


(b) $n = 100$

Figure 4.1: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the factors number k with $k_j = 3$ fixed on the data sets of a 7-dimensional x ($d = 7$) generated from a 3-dimensional y ($k = 3$) with different sample sizes for NFA.



(a) $n = 20$



(b) $n = 100$

Figure 4.2: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the gaussians number k_j with $k = 3$ fixed on the data sets of a 7-dimensional x ($d = 7$) generated from a 3-dimensional y ($k = 3$) with different sample sizes for NFA.

Table 4.2: Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different sample sizes for selecting gaussians number k_j with $k = 3$ fixed for NFA in 50 experiments

criteria	$n = 20$			$n = 40$			$n = 100$		
	U	S	O	U	S	O	U	S	O
AIC	9	31	10	4	34	12	1	41	8
BIC	22	26	2	17	32	1	5	43	2
CAIC	28	22	0	21	29	0	15	35	0
10-fold CV	6	29	15	3	32	15	0	39	11
BYY	7	31	12	4	37	9	1	44	5

4.3.2 Effects of Data Dimension on Model Selection Criteria

Next we investigate the effect of data dimension on each criterion. The dimension of y is $k = 3$, the noise covariance matrix Σ_2 is equal to $0.1I$, and the sample size is $n = 80$. The dimension of x is $d = 5$, $d = 10$, and $d = 20$. The results with different factors number k and fixing $k_j = 3$ are shown in Figure 4.3. The results with different k_j and fixing $k = 3$ are shown in Figure 4.4. Table 4.3 and Table 4.4 illustrate the numbers of underestimating, success, and overestimating of each method for selecting k and k_j respectively in 50 experiments.

When the dimension of x is 5, we observe that all these criteria tend to select the right hidden factors number 3. However, when the dimension of x is increased to 20, BYY, 10-fold CV get the right number 3, but CAIC and BIC tend to underestimate the hidden factors number and AIC tend to overestimate the hidden factors number. Similar observations can be obtained

Table 4.3: Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different data dimensions for selecting hidden factors number k with $k_j = 3$ fixed for NFA in 50 experiments

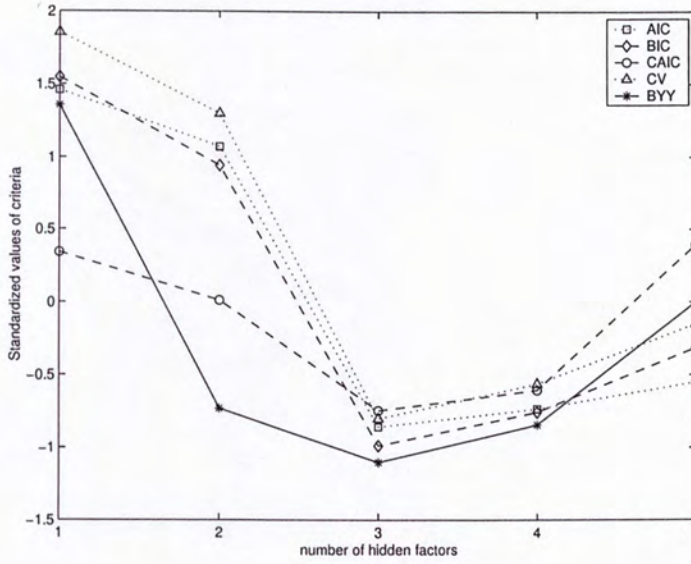
criteria	$d = 5$			$d = 10$			$d = 20$		
	U	S	O	U	S	O	U	S	O
AIC	2	42	8	0	39	11	0	36	14
BIC	3	47	0	10	40	0	13	34	3
CAIC	4	46	0	13	37	0	20	29	1
10-fold CV	0	40	10	0	40	10	0	39	11
BYY	0	47	3	1	44	5	2	40	8

Table 4.4: Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different data dimensions for selecting gaussians number k_j with $k = 3$ fixed for NFA in 50 experiments

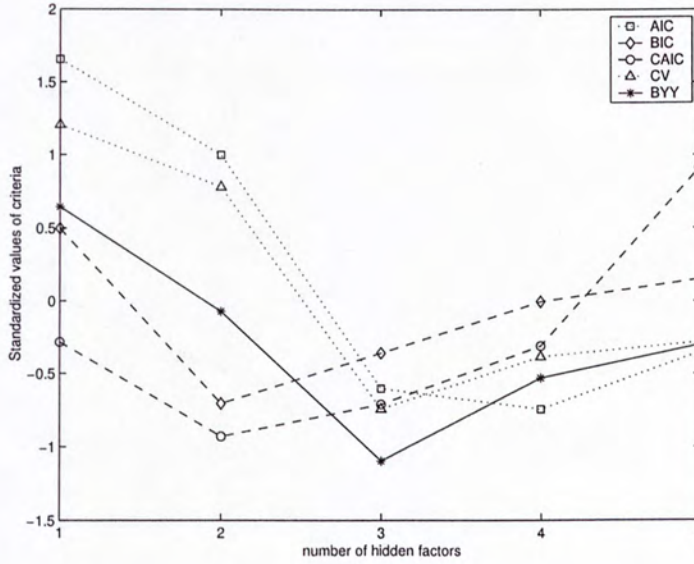
criteria	$d = 5$			$d = 10$			$d = 20$		
	U	S	O	U	S	O	U	S	O
AIC	0	40	10	2	38	10	1	37	12
BIC	5	43	2	11	38	1	13	33	4
CAIC	10	40	0	14	35	1	22	24	4
10-fold CV	1	39	10	0	39	11	0	40	10
BYY	3	43	4	4	37	9	5	32	13

in Table 4.3.

When the dimension of x is 20, AIC and 10-fold CV get the right gaussians number 3 of k_j , BIC and CAIC tend to underestimation while BYY criterion has a risk of overestimation. Similar observations can be obtained in Table 4.4.

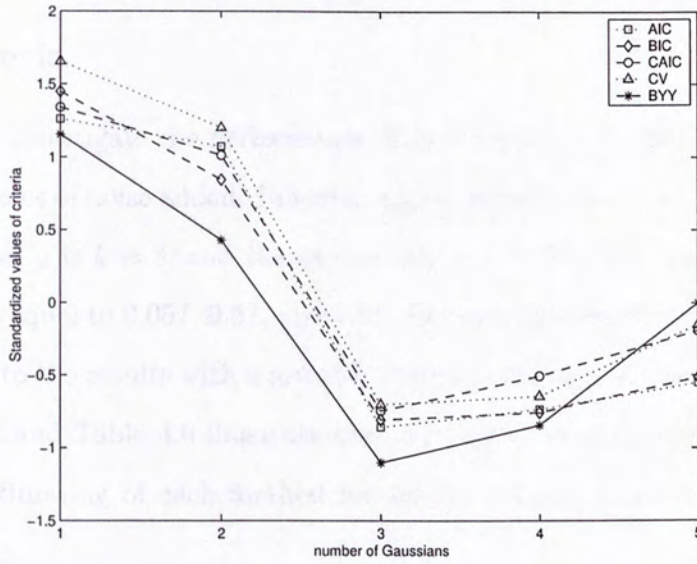


(a) $d = 5$

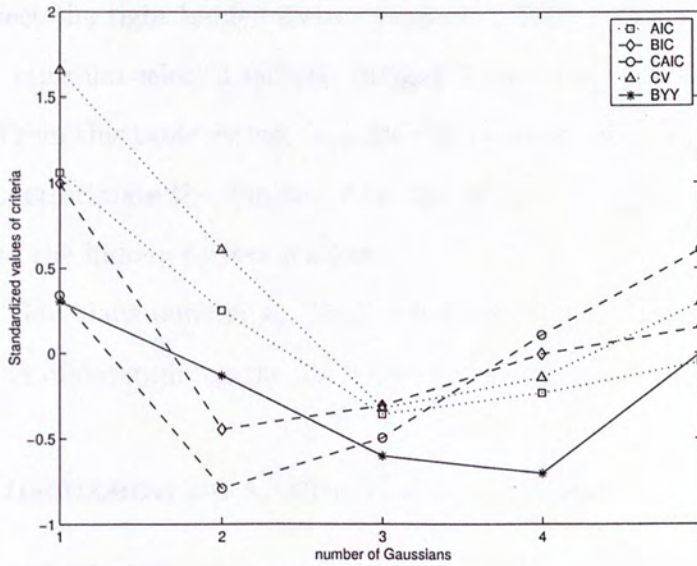


(b) $d = 20$

Figure 4.3: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the factors number k with fixing $k_j = 3$ on the data sets of a x with different dimensions generated from a 3-dimensional y ($k = 3$) for NFA.



(a) $d = 5$



(b) $d = 20$

Figure 4.4: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the gaussians number k_j with fixing $k = 3$ on the data sets of a x with different dimensions generated from a 3-dimensional y ($k = 3$) for NFA.

4.3.3 Effects of Noise Variance on Model Selection Criteria

We further investigate the performance of each criterion on the data sets with different scales of noise added. In this example, the dimension of x is $d = 7$, the dimension of y is $k = 3$, and the sample size is $n = 80$. The noise covariance matrix Σ is equal to $0.05I$, $0.5I$, and $1.5I$. Because the results with different k_j are similar to the results with number k , they are shown in the same Figure 4.5. Table 4.5 and Table 4.6 illustrate the numbers of underestimating, success, and overestimating of each method for selecting k and k_j respectively in 50 experiments.

When the noise covariance matrix is $1.5I$, we see that only AIC and 10-fold CV select the right hidden factors number 3, BIC, CAIC select 2 factors while BYY criterion select 4 factors. Similar observations can be observed in Table 4.5. From this table we can find, for a large noise variance, CAIC is high likely to underestimate the number, AIC and 10-fold CV have a slight risk of overestimate the hidden factors number.

For the Gaussians number k_j , Table 4.6 shows that the results are similar to the results of determining the hidden factors number k .

4.3.4 Discussion on Computational Cost

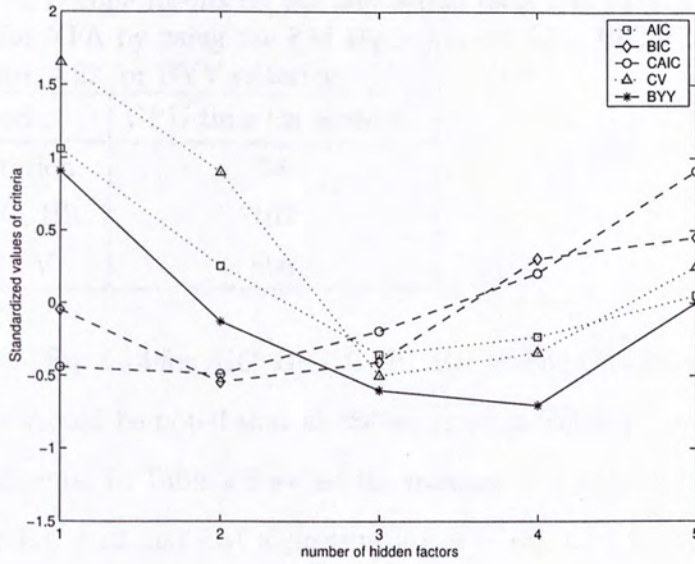
All the experiments were carried out using MATLAB R12.1 v.6.1 on a P4 2GHz 512KB RAM PC. We illustrate the computational results in Table 4.7 for the experiment to determine the hidden factors number k in the first example described in subsection 4.3.1 with sample size $n = 40$ by using the EM algorithm

Table 4.5: Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different noise variances for selecting hidden factors number k with fixing $k_j = 3$ for NFA in 50 experiments

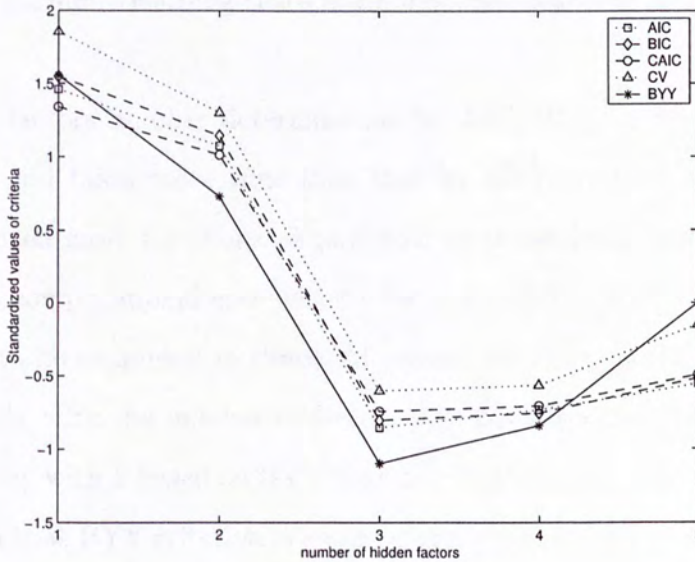
criteria	$\Sigma = 0.05I$			$\Sigma = 0.5I$			$\Sigma = 1.5I$		
	U	S	O	U	S	O	U	S	O
AIC	0	41	9	1	41	8	1	37	12
BIC	3	47	0	4	45	1	15	28	5
CAIC	5	45	0	6	44	0	25	21	4
10-fold CV	0	40	10	0	41	9	1	39	10
BYY	0	47	3	3	43	4	10	26	14

Table 4.6: Numbers of underestimating (U), success (S), and overestimating (O) by each criterion on the data sets with different noise variances for selecting gaussians number k_j with fixing $k = 3$ for NFA in 50 experiments

criteria	$\Sigma = 0.05I$			$\Sigma = 0.5I$			$\Sigma = 1.5I$		
	U	S	O	U	S	O	U	S	O
AIC	0	40	10	1	39	10	2	38	10
BIC	6	43	1	7	43	0	20	25	5
CAIC	10	40	0	9	41	0	29	21	0
10-fold CV	2	38	10	1	39	10	0	40	10
BYY	0	45	5	2	43	5	8	26	16



(a) $\Sigma = 1.5I$



(b) $\Sigma = 0.05I$

Figure 4.5: The curves obtained by the criteria AIC, BIC, CAIC, 10-fold CV and BYY for selecting the factors number k and the gaussians number k_j on the data sets of a 7-dimensional x ($d = 7$) generated from a 3-dimensional y ($k = 3$) with different noise variances for NFA.

Table 4.7: CPU time results on the simulation data sets with $n = 40$, $d = 7$, and $k = 3$ for NFA by using the EM algorithm for AIC, BIC, CAIC and CV, algorithm Eq. 4.22 for BYY criterion

method	CPU time (in seconds)
BYY criterion	58
AIC, CAIC, BIC	103
10-fold CV	698

from Eq. 4.7 - Eq. 4.14 for AIC, BIC, CAIC and 10-fold CV, Eq. 4.22 for BYY criterion. It should be noted that all values given in Table 4.7 are the average of 50 experiments. In Table 4.8 we list the running time of both BYY harmony learning by Eq. 4.22 and EM algorithm Eq. 4.7- Eq. 4.14 when $k = 3$ using the same data as sect. 4.3.1. This table shows that the the algorithm derived from BYY harmony learning takes much less time than the conventional EM algorithm.

Hidden factors number determination by AIC, BIC, CAIC takes similar CPU time and takes more time than that by BYY criterion since the EM algorithm need more iterations. The m -fold cross-validation method requires the highest computational cost because for each candidate model the parameters have to be estimated m times. Moreover, the computational cost grows exponentially with the number of factors k in EM algorithm but empirically grows linearly with k based on BYY harmony learning [40]. Since experiments have shown that BYY criterion is superior or comparable to typical statistical model selection criteria, corresponding BYY automatic model selection algorithm is less computationally intensive such that BYY harmony learning are considered more favorable for non-Gaussian factor analysis.

Table 4.8: CPU time results on the simulation data sets with $n = 40$, $d = 7$, and $k = 3$ for NFA by using the EM algorithm and the algorithm derived from BYY harmony learning where set the candidate $k = 3$

method	CPU time (in seconds)
BYY harmony	13
EM algorithm	31

4.4 Summary

We have made an experimental comparison on several typical model selection criteria by using them to determine hidden factors number and the Gaussians number for each factor. The considered criteria include four typical model selection criteria AIC, BIC, CAIC, 10-fold CV, and the model selection criterion obtained from BYY harmony learning, namely BYY criterion. We observe that BYY criterion is superior or comparable to other methods and has high successful rate in most cases. BIC also got a high successful rate when the data dimension is not too high. CAIC has an underestimation tendency while AIC and 10-fold CV have an overestimation tendency. The cross-validation method requires a highest computing cost.

Chapter 5

Conclusions

In this thesis, we first analyze the experiment results of two architectures of BYY harmony learning with automatic model selection during parameter estimation in implementation of BFA. Based on the analysis, we proposed two heuristic methods and a combination strategy which can efficiently improve the correct rate for implementing BFA with model selection done automatically.

Secondly, we have made experimental comparisons on determining the hidden factor number in implementing BFA. The methods include four typical model selection criteria AIC, BIC, CAIC, 10-fold CV, and BYY criterion as well as BYY learning with automatic model selection. We have observed that the performances by BYY criterion and BYY-AUTO are either superior or comparable to other methods in most cases. Both BYY criterion and BYY-AUTO have high successful rates except the cases of large noise variance. BIC also got a high successful rate when the data dimension is not too high and the noise variance is not too large. CAIC has an underestimation tendency while AIC and 10-fold CV have an overestimation tendency. Moreover, BYY-AUTO needs a much less computing time than all the considered criteria including

BYY criterion.

Thirdly, we have made an experimental comparison on these typical model selection criteria and BYY criterion by using them to determine hidden factors number and the Gaussians number for each factor for NFA. We observe that BYY criterion is superior or comparable to other methods and has high successful rate in most cases. BIC also got a high successful rate when the data dimension is not too high. CAIC has an underestimation tendency while AIC and 10-fold CV have an overestimation tendency. The cross-validation method requires a highest computing cost. Therefore, BYY harmony learning is a more preferred tool for the model selection in BFA and NFA.

Bibliography

- [1] Akaike, H., A new look at statistical model identification. *IEEE Transactions on Automatic Control* , **19**:716-723, 1974
- [2] Akaike, H., Factor analysis and AIC. *Psychometrika*, **52(3)**:317-332, 1987
- [3] Anderson, T.W. and Rubin, H., Statistical inference in factor analysis. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, **5**:111-150, 1956
- [4] Attias, H., Independent factor analysis. *Neural Computation*, **11**: 803-851, 1999.
- [5] Bartholomew, D.J. and Knott, M., Latent variable models and factor analysis. *Kendall's Library of Statistics*, Oxford University Press, New York, **7**, 1999
- [6] Barron, A. and Rissanen, J., The minimum description length principle in coding and modeling. *IEEE Trans. Information Theory*, **44**:2743-2760, 1998

- [7] Belouchrani, A. and Cardoso, J. Maximum likelihood source separation by the expectation-maximization technique: deterministic and stochastic implementation. *Proc. NOLTA95*, 49-53, 1995
- [8] Bertin, E. and Arnouts, S., Extractor: Software for source extraction. *Astron. Astrophys. Suppl. Ser.*, **117**: 393-404, 1996
- [9] Boulard, H. and Kamp, Y., Auto-association by multilayer perceptrons and sigular value decomposition. *Biol. Cybernet*, **59**:291-294, 1988
- [10] Bozdogan, H., Model selection and Akaike's information criterion (AIC): the general theory and its analytical extensions. *Psychometrika*, **52(3)**:345-370, 1987
- [11] Belouchrani, A. and Cardoso, J.F., Maximum likelihood source separation by the expectation-maximization technique: Deterministic and stochastic implementation. *Proc. NOLTA95*, 49-53, 1995
- [12] Cardoso, J.F., Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. *Proc.ICASSP97*,3617-3620, 1997
- [13] Cattell, R., The score test for the number of factors. *Multivariate Behavioral Research*, **1**:245-276, 1966
- [14] Cichocki, A. and Amari, S.I., Adaptive Blind Signal and Image Processing. New York: Wiley, 2002.
- [15] Dayan, P. and Zemel, R.S., Competition and multiple cause models. *Neural Computation*, **7**:565-579, 1995

- [16] Heinen, T., Latent class and discrete latent trait models: Similarities and differences. Thousand Oaks, California: Sage, 1996
- [17] Hyvarinen, A., Independent component analysis in the presence of Gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22: 49-67, 1998.
- [18] Hyvarinen, A., Karhunen, J. and Oja, E., Independent Component Analysis. New York: Wiley, 2001
- [19] Kaiser, H., A second generation little jiffy. *Psychometrika*, 35:401-415, 1970
- [20] Moulines, E., Cardoso, J. and Gassiat, E., Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. *Proc. ICASSP97*, 3617-3620, 1997
- [21] Rissanen, J., Modeling by shortest data description. *Automatica*, 14:465-471, 1978
- [22] Rubin, D. and Thayer, D., EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69-76, 1982
- [23] Sahami, M., Hearst, M., and Saund, E., Applying the Multiple Cause Mixture Model to Text Categorization . *Proceedings of ICML-96, 13th International Conference on Machine Learning*, pages 435-443, San Francisco, CA, 1996
- [24] Saund, E., A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7:51-71, 1995

- [25] Schwarz, G., Estimating the dimension of a model. *The Annals of Statistics*, **6(2)**:461-464, 1978
- [26] Sclove, S.L., Some aspects of model-selection criteria. *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, Kluwer Academic Publishers, Dordrecht, the Netherlands, **2**:37-67, 1994
- [27] Stone, M., Use of cross-validation for the choice and assessment of a prediction function. *Journal of the Royal Statistical Society*, **B 36**:111-147, 1974
- [28] Treier, S. and Jackman, S., Beyond factor analysis: modern tools for social measurement. Presented at the 2002 Annual Meetings of the Western Political Science Association and the Midwest Political Science Association, 2002
- [29] Xu, L., Least mean square error reconstruction for self-organizing neural-nets. *Neural Networks*, **6**:627-648, 1993
- [30] Xu, L., Bayesian-Kullback coupled Ying-Yang machines: Unified learnings and new results on vector quantization. *Proc. Intl. Conf. on Neural Information Processing (ICONIP95)*, Beijing, China, 977-988, 1995
- [31] Xu, L., Yang, H. H. and Amari, S.I., Signal source separation by mixtures: accumulative distribution functions or mixture of bell-shape density distribution functions. *Resentation at FRONTIER FORUM. Japan: Institute of Physical and Chemical Research*, April, 1996

- [32] Xu, L., Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach (III): Models and Algorithms for Dependence Reduction, Data Dimension Reduction, ICA and Supervised Learning. *in K.M. Wong, et al, eds, Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective*, Springer-Verlag, 43-60, 1997
- [33] Xu, L., Bayesian Kullback Ying-Yang Dependence Reduction Theory. *Neurocomputing*, **22(1-3)**:81-112, 1998
- [34] Xu, L., Bayesian Ying-Yang learning theory for data dimension reduction and determination. *Journal of Computational Intelligence in Finance*, **6(5)**:6-18, 1998
- [35] Xu, L., Temporal BYY learning for state space approach, hidden markov model and blind source separation. *IEEE Trans on Signal Processing*, **48**:2132-2144, 2000
- [36] Xu, L., BYY harmony learning, independent state space, and generalized APT financial analyses. *IEEE Transactions on Neural Networks*, **12(4)**:822-849, 2001
- [37] Xu, L., BYY harmony learning, structural RPCL, and topological self-organizing on mixture models. *Neural Networks*, **15**:1125-1151, 2002
- [38] Xu, L., Independent component analysis and extensions with noise and time: A Bayesian Ying-Yang learning perspective. *Neural Information Processing Letters and Reviews*, **1(1)**:1-52, 2003

- [39] Xu, L., BYY learning, regularized implementation, and model selection on modular networks with one hidden layer of binary units. *Neurocomputing*, **51**:277-301, 2003
- [40] Xu, L., Advances on BYY Harmony Learning: Information Theoretic Perspective, Generalized Projection Geometry, and Independent Factor Autodetermination. *IEEE Transactions on Neural Networks*, **15**(4):885-902, 2004.
- [41] Xu, L., Bi-directional BYY learning for mining structures with projected polyhedra and topological map. *Proceedings of IEEE ICDM2004 Workshop on Foundations of Data Mining*, Brighton, UK, Nov. 1-4, T.Y. Lin, S. Smale, T. Poggio, and C.J.Liau, eds, 2-14, 2004.
- [42] Z.Y. Liu, K.C. Chiu, and L. Xu, Investigations on Non-Gaussian Factor Analysis. *IEEE Signal Processing Letters*, **11**(7):597-600, 2004.

CUHK Libraries



004278869