

# A self-adaptive multi-view framework for multi-source information service in cloud ITS

Shan Xue<sup>1,2</sup> · Li Xiong<sup>1</sup> · Shufen Yang<sup>1</sup> · Lu Zhao<sup>1</sup>

Received: 29 April 2015 / Accepted: 31 August 2015 / Published online: 16 September 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** In traditional intelligent transportation system (ITS), the information source is often collected from a single view. However, as ITS becoming increasingly complicated, there is a need to describe one object from several different information views/domains. Information from multiple sources can provide extra worthwhile information for ITS users especially in cloud environment. Moreover, existing local ITSs usually provide information by fixed algorithms, which is not adequate to the dynamic transportation scenarios that produce big traffic data with time series. In this paper, we propose a complete self-adaptive multi-view framework for multi-source information service in cloud ITS, which mainly consists of a Newton multi-parameter optimization, a multi-layer feed-forward neural network and a finite multi-view mixture distribution. A simulation on real-world application, with six different types of information views, demonstrates the underlying effectiveness of the proposed framework.

**Keywords** Intelligent transportation system · ITS · Multi-source information · Multi-layer feed-forward neural network · Self-adaption · Cloud computing

## 1 Introduction

The intelligent transportation system (ITS) is one of advanced applications, which without embodying intelligence as such, aims to provide innovative services relating to different modes of transportation managements and to enable various users to be better informed and make safer, more coordinated and smarter use of transportation networks (Leviäkangas et al. 2007; Wang 2010; Zhang et al. 2011; Merzouki et al. 2013; Dinakaran 2014). For example, Stilwell and Bay (1993) propose a self-adapt simulation theory on ant colony optimization (ACO) for ITS. Barceló et al. (2005) regard ITS as typical dynamics and high sensitive features by reviewing the design, analysis and evaluation methods of ITS. Besides, researchers of ITS focus on traffic information collections and processings (Wang et al. 2005; Delot et al. 2011).

Despite of the popularity of ITS, researchers recently find that using information from a single view is inadequate to make a full use of the functions of complicated transportation systems (El Faouzi et al. 2011; Lim et al. 2014; Kramers 2014). The idea of multi-view information service is mainly developed from the multi-source information process, such as unstructured and structured information (Wu et al. 2014, 2015a). Yager (2005) puts forward a framework of multi-source information integrations based on the judgements of information sharing conflicts. This framework is used to discuss the rationality of any information from other resources than which can be obtained in the processes of information integrations. Su (2008) raises

---

✉ Li Xiong  
xiongli8@shu.edu.cn  
Shan Xue  
xueshan@shu.edu.cn  
Shufen Yang  
13720521@shu.edu.cn  
Lu Zhao  
lulukeerle@shu.edu.cn

<sup>1</sup> School of Management, Shanghai University, Baoshan District, Shanghai, China

<sup>2</sup> Lab of Decision Systems and e-Service Intelligence, Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Sydney, Australia

a new Bayes optimal classifier based on fuzzy information sets embedded according to the evolutionary process of fuzzy theory. From the critical characteristics of uncertain multi-source information on the both positive and negative aspects, an artificial neural network (ANN) method based on knowledge learning is proposed which aims to realize the automatic Bayes optimal classifier. However, Ali et al. (2012) believe through effective information integration from multiple sources, all kinds of information can be integrated and be complementary formed to meet the users' needs either accurately or conveniently. That is a important trend for information services in the future.

At present, different information sources need effective integration and complementarity as well as to provide accurate and convenient information services for users in ITS (Ma et al. 2012; Mugellini et al. 2013). For example, Shanghai, as one of innovation pilot demonstration cities on cloud computing service in China (Hao et al. 2012), is promoting multi-source intelligent projects in three areas which are parts of Shanghai's Smart City Strategy 2011–2013 (Shanghai Government 2011; Zhou 2014). The three areas include cloud computing, networking and other information service systems to support the constructions of intelligent transport management. In face of the shortages of limited traffic data from traditional sources and disorganized data from new media, we need to change the mode of multi-source information service and base on a novel ITS framework to deal with and make use of the challenge of cloud computing (Wan et al. 2014).

As a multi-view framework can provide more useful information than a single view (Hoffmann et al. 2008), multiple information sources can be potentially useful for improving existing performance of ITSs. Existing ITSs usually provide information by fixed algorithms (Merzouki et al. 2013), which is not adequate to the dynamic transportation scenarios. It is desired to design a self-adaptive multi-view framework for multi-source information service in cloud ITS, which aims to increase its efficiency and reduce complexity of ITSs.

Neural Network (NN), which is short of ANN, adjusts the weights between neurons by learning under certain topology rules. According to Li et al. (2004), the self-adapt learning process of NN is optimized at the same time with the network topology structure optimization so as to improve the network adaptability. However, most NN application on ITS focus on single view predictions. Liu et al. (2012) propose a hybrid model based on NN for traffic prediction which is one of the most important applications of ITS. Colombaroni and Fusco (2014) deal with the application of NNs to model car-drivers behaviors. This study shows NNs provide a good approximation of driving patterns, so that NNs can be suitably implemented

in microsimulation models which represent individual of the driver population through systematic observations.

The advances of cloud computing and its platform have provided a promising opportunity to resolve the increasing transportation problems (Ashokkumar et al. 2015). Cloud intelligent transportation systems, which set ITS in cloud environment and use cloud computing techniques, can provide services such as autonomy, mobility, decision support, the standard development environment for traffic management strategies, and so on (Trivedi et al. 2012). Based on mobile multi-agent technology, Li et al. (2011) embed cloud computing in agent-based traffic management systems to cope with the large amounts of storage and computing resources which are required to use traffic strategy agents and mass transport data effectively. For a intelligent carpool system (ICS), which provides carpoolers the use of the carpool services via a smart hand-held device at anywhere and at any time, Huang et al. (2015) propose a genetic-based carpool route and match this algorithm in cloud environment. Research by Ramesh et al. (2013) indicate cloud is the best platform to implement ITS service. In such a platform, the number of passengers in a bus stop can be calculated and the bus service can be regulated depending on the passenger's arrivals.

Motivated by the above discussions, we propose a self-adaptive multi-view framework for multi-source information service in cloud ITS, which mainly consists of a Newton multi-parameter optimization, a MFL neural network and a finite multi-view mixture distribution. A simulation on real-world application [i.e., transportation datasets from Shanghai Statistical Yearbook (2001–2014), with six different types of information views], demonstrates the underlying effectiveness of the proposed framework.

The rest of the paper is organized as follows. Section 2 introduces the related works of ITS, cloud computing with big data and ANN. In Sect. 3, a self-adaptive multi-view framework for multi-source information service in cloud ITS is proposed and described. In Sect. 4, we carry out a simulation and demonstrate the self-adaptive model. Conclusions and future study plan are presented in Sect. 5.

## 1.1 Our contributions

To meet the needs of information service in ITS, especially in cloud and big data environment, and overcome the challenges from existing ITS service frameworks, we develop a new self-adaptive multi-view framework to support multi-source information services in cloud data-driven ITS, where the great contributions are summarized as follows:

- Running in a data-driven ITS particular in cloud and big data environment, a global framework is designed for information decision support, which processes raw data from multiple sources with classifications of real-time data and outdated data (see Sect. 3);
- As a key module in our proposed framework, we define a local process of Newton iterative method, which brings better effect than the general multi-parameter optimization for NN (see Sect. 3.2);
- To achieve self-adaption on information services, we introduce multi-layer feed-forward (MLF) neural network to train datasets from data-driven ITS for decision support purpose (see Sects. 3.3 and 4.2);
- For prediction, finite multi-view mixture distribution offers us a possibility in distribution of heterogeneous data structure with a more reasonable explanation and a more flexible prediction (see Sects. 3.4 and 4.3);
- The self-adaptive multi-view framework for multi-source information service in cloud ITS is simulated for Newton iterative method, MLF neural network and finite mixture distribution, which both show remarkable results on Shanghai traffic datasets (2001–2014) (see Sect. 4).

This paper presents latest study and results of our cloud ITS framework, and has a major new focus on the intelligent transportation techniques for social information service. The proposed cloud ITS framework is also a key part of smart city strategies in big data era. The paper covers a breakthrough on integrations of Newton method for multi-parameter optimization, MLF neural network for multi-source datasets training, and finite mixture distribution for multi-view distribution.

## 2 Related work

In this section, we briefly review the related work which can be grouped into three categories. The first category is about ITS especially in information service domain. The second category is cloud computing with big data network which is integrated in data-driven systems. The third category focuses on ANNs which is a key tool to achieve self-adaption and intelligence in cloud ITS.

### 2.1 Intelligent transportation system

With the widespread adoption of location tracking technologies like GPS, the domain of intelligent transportation services has seen growing interests in the last few years. For example, the bus network design is known to be a complex, nonlinear, non-convex, multi-objective NP-hard problem. Chen et al. (2014) trace taxi GPS to acquire human mobility patterns. Through analyzing the pick-

up/drop-off densities of taxi passengers, hot areas of public transportation are clustered and candidate bus stops are proposed based on the analyses. Similarly, Tao (2007) propose a practical and applicable taxi-sharing system based on the use of ITS technologies. This work has been developed in Taipei city which is easy for members to use and inexpensive for the service provider to operate. By dynamic rideshare matching processes based on the Internet and wireless communication network infrastructure is embedded.

A public bus transportation system consists of RFID module, in-bus module, base station module and bus stop module. In such a system, a per-stop statistical analysis is carried out based on the number of passengers and a recommendation report, and multiple control-point strategy for holding control of a bus transit system is presented by Koehler et al. (2011). The model developed is deterministic and assumes the availability of real-time information and historical data from the system.

Information services in ITSs should combine the real-time location-based data from multiple sources, and provide useful information to end-users (Maleki-Dizaji et al. 2014). These data may be consisted of static or dynamic location based data, Biem et al. (2010) suggest the data should be collected from other related sources as well, e.g., weather, video cameras, etc. Multi-source data collection enables real-time traffic monitoring and management with a broader scope and sustainability than usually achieved. Despite the challenge of real-time analysis in ITSs, the computing infrastructure requires to support the special ITS capabilities, especially on large data volumes, heterogeneous data structure and multiple data sources such as from government agencies, commercial enterprises and end-user commuters.

### 2.2 Cloud computing with big data

According to the National Institute of Standards and Technology (NIST) (Mell and Grance 2011), cloud computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The NIST's definition also describes the following essential characteristics for cloud computing, which are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. As cloud service models consist of the Cloud Software as a Service (SaaS), the Cloud Platform as a Service (PaaS), and the Cloud Infrastructure as a Service (IaaS), cloud federation can allow individual cloud providers working

collaboratively to offer best-effort services to customers (Liu et al. 2011).

In our framework of cloud ITS self-adaptive multi-view information service, we expect users obtain computing capabilities (Saukh et al. 2014; Zhang et al. 2015) as required with the cloud provider but without human interaction. Branch et al. (2014) suggest that cloud computing and big data techniques require network architectures which differ from the traditional client-server applications. The changes in networks of information flow direction and unpredictable bursts of data, have attributed to the evolutions of network architectures, and traditional client-server do not meet the requirements of cloud computing and big data applications (Duan et al. 2012). Introducing cloud computing in Branch et al. (2014) to our cloud ITS self-adaptive multi-view information service framework, the users can be accessible over a network where allow access via numerous heterogeneous client platforms. In such a network structure, the resources will be locational independent but invisible in cloud. Meanwhile, the cloud providers will combine these multiple resources to serve numerous cloud users dynamically by users' demands, such as different physical and virtual resources. Furthermore, cloud services and resources will be automatically controlled and optimized by a cloud system.

To address these challenges of cloud services especially in multi-source information self-adaption requirements, we focus on information flow between each nodes in cloud ITS information services networks, which include information providers, end-uses, and platforms. Dikaiakos et al. (2009) indicate that the technology of cloud computing must be developed to work in practice. Although several companies have already built Internet consumer services that use cloud computing infrastructure, such as search, social networking, Web email and online commerce, deep applications of cloud services is still facing many of the challenges which need us to find a solution (Leu et al. 2013). Other issues for cloud ITS information services in industry are also vital enough to take ITS into considerations, e.g., real-time and dynamics (Yang et al. 2012), security (Subashini and Kavitha 2011). Zhang et al. (2010) have surveyed the state-of-the-art of cloud computing, covering its essential concepts, architectural designs, prominent characteristics, key technologies as well as research directions. The result shows that despite the significant benefits offered by cloud computing, the current technologies are not matured enough to realize its full potential.

### 2.3 Artificial neural network

Self-adaptive strategies, e.g., differential evolution (DE) (Wu and Cai 2014), artificial immune system (AIS) (Wu

et al. 2013a, b), and different kinds of artificial neural network (ANN) (Jain et al. 1996; Balcazar et al. 1997; Cireşan et al. 2012), etc., are well used in data mining and machine learning areas, such as Bayesian Networks (Wu et al. 2015b).

In computer science related fields, ANNs are inspired by animal central nervous systems, in particular the brain, which are usually presented as systems of interconnected neurons that can compute values from inputs by feeding information through the network (Rigatos et al. 2013). A feed-forward NN is an ANN where connections between the units do not form a directed cycle (Sanger 1989). As the learning speed of feed-forward NNs is in general far slower than required, a learning algorithm for single hidden layer feed-forward neural networks (SLFNs) which randomly chooses the input weights and analytically determines the output weights of SLFNs is presented (Huang et al. 2004).

In ITS application domain, Xie et al. (2014) forecast the short-term passengers' flow on high-speed railway with neural networks. Flow forecasting of short-term passengers is an important component of transportation systems and the forecasting results can be applied to support ITS operations, such as planning, revenue management and offer real-time information services. In that research, the amount of passengers who arrive at each station or depart from each station are obtained to historical data of passenger flows. Meanwhile, the forecasting algorithm is based on a single-hidden layer feed-forward neural network with Back-propagation. However the training set ignores special issues, such as the holiday/weekends trends and conventional forecasting.

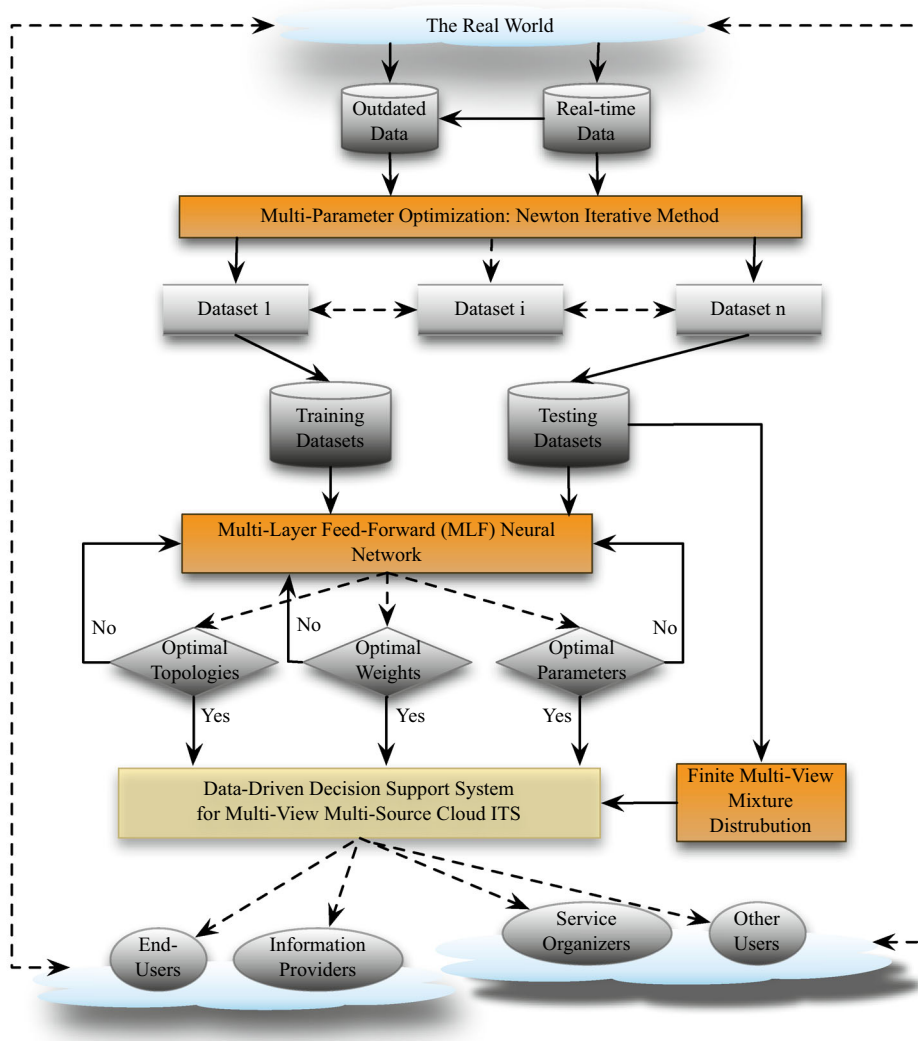
To the best of our knowledge, our work is the first to apply the MLF neural network to multi-source information service in cloud ITS.

## 3 A self-adaptive multi-view framework for multi-source information service in cloud ITS

### 3.1 Framework

Facing complex urban traffic problems, we chose to make a full use of real world data referring to multi-views under an overall cloud ITS framework. Motivated by data-driven systems, we desire to achieve self-adaption both in data processes and information presentations. In this paper, as shown in Fig. 1, we propose a self-adaptive multi-view framework for multi-source information service in cloud ITS. In this framework, all data including training datasets and testing datasets are from multiple real-world social information sources, without pre-process and tagged with

**Fig. 1** A self-adaptive multi-view framework for multi-source information service in cloud ITS



time label. Thus, raw data we used in cloud ITS are typically disorganized.

The core processes in this framework lie on multi-parameter optimization where we use Newton iterative method (see Sect. 3.2), multi-layer feed-forward neural network (see Sect. 3.3) and finite multi-view mixture distribution (see Sect. 3.4). The detailed processes of this framework are as follows.

*Step 1 Raw data classification* Raw data from real world are input into the cloud ITS. All raw data should be classified by time labels, so data in this system are separated as real-time data and outdated data.

*Step 2 Multi-parameter optimization: Newton iterative method* Both real-time data and outdated data will be input into and be pre-processed by the multi-parameter optimization module. We choose Newton iterative method to achieve multi-parameter optimization for its marked

advantages. Previous dataset from outdated data will be output as dataset 1 to dataset  $i$  and the same to real-time data as dataset  $i + 1$  to dataset  $n$ . The training datasets for next training module consist of all outdated datasets and part of real-time datasets. The rest of real-time datasets will be included to testing datasets.

*Step 3 Multi-layer feed-forward (MLF) neural network training* We will then send training datasets and testing datasets to multi-layer feed-forward (MLF) neural network where MLF neural network will be trained by training datasets. In this framework, NN strategies like optimal topologies, optimal weights and optimal parameters are embedded. For evolutionary training, loops end only if testing results are optimal topologies, optimal weights or optimal parameters, else MLF neural network continues training.

*Step 4 Finite multi-view mixture distribution* Testing datasets in this system will be processed by finite multi-

view mixture distributions. First of all, data will be labeled by finite views. Then, each view will be weighted by its importance in specific settings. In the end, finite multi-view mixture distribution will be constructed by distributions of single views and their weights.

*Step 5 Data-driven decision support system for multi-view multi-source cloud ITS* Input of data-driven decision support system for multi-view multi-source cloud ITS. The inputs consists of outputs from MLF neural network and finite multi-view mixture distributions. This module will not be detailedly discussed in this paper.

*Step 6 Framework circulation* The users of this multi-view multi-source cloud ITS framework which are also considered as users of its data-driven decision support system may be ITS end individuals, ITS information providers, ITS service organizers and other users of related applications. All data produced by these users will be collected into our framework and continuously improve system's functions.

The framework above reflects the dynamic characteristics of real-time information self-adapt transfer learning processes. The adaptive characteristics of this framework are shown as following:

- Dimension reduction is realized by multi-parameter optimization in Newton iterative method;
- The cloud ITS achieves a optimal neural network topology, that is, we should obtain optimal weights between layers and optimal iteration parameters before determining the logical structure of the hidden layer; and
- Through the positive output data of MLF neural network, flexible global distribution is fitted by finite multi-view mixture distribution.

### 3.2 Multi-parameter optimization: Newton iterative method

In this paper, we introduce multi-parameter optimization to find minimum value of function  $S$  to effectively reduce the dimensions of training samples and testing samples. In the multi-source information NN model which provides self-adaptive service, the dimension of experimental sample  $d$  is extremely high, and the clusters of minimum values are always in the multi-dimensional spaces. Therefore,  $S$  is used as a scoring function for meta parameter vector  $\theta$  in  $d$  dimensions. According to Matthew (2012), if local minimum value can be found in preprocessed multi-source data, the non-minimum value spaces will be eliminated.

The general local process of multi-parameter optimization is defined as,

$$\theta^{i+1} = \theta^i + \lambda^i v^i, \quad (1)$$

where  $\theta^i$  is an estimated parameter of iteration step  $i$ ,  $v^i$  is the  $d$ -dimensional vector referring to the direction of movement in the next iteration.

Theoretically, through finite iterations,  $\theta^S$ , which corresponds to  $S_{min}$ , can be found. While feed-forward NN usually uses the steepest descent algorithm, the steepest decline does not necessarily point to the minimum gradient theory. Thus, the general multi-parameter optimization process of local iteration is not the preferred iteration in this study. Instead we chose Newton iterative method as a local process, which brought better effect than the general one (Sherman 1978). The details of this Newton iterative method are shown in Algorithm 1.

---

#### Algorithm 1 Multi-parameter optimization: Newton iterative method

---

**Input:**

Parameter vector  $\theta$ ;  
Weight function  $S$ .

**Output:**

Parameter vector  $\theta^S$  with  $S_{min}$ .

// Initialization

```

1:  $\theta^0 \leftarrow$  Iteration step 0 on parameter vector;
2:  $\epsilon \leftarrow$  Bounded variable.
3: while  $S(\theta^{i+1}) \leq S(\theta^i)$  do
4:    $\theta^{i+1} \leftarrow$  via Eq. (4);
5:    $S(\theta^{i+1}) \leftarrow$  Function  $S$  on  $\theta^{i+1}$  via Eq. (2);
6:    $g(\theta^{i+1}) \leftarrow$  The first derivative of  $S(\theta^{i+1})$ ;
   // Calculation for matrix  $H^{-1}$ 
7:   for all  $i + 1 \leq l$  and  $m \leq d$  do
8:      $h_{l,m} \leftarrow$  Elements of matrix  $H(\theta^{i+1}) \leftarrow$  via Eq. (3);
9:      $H^{-1}(\theta^{i+1}) \leftarrow$  The inverse matrix of  $H(\theta^{i+1})$ ;
10:  end for
   // Update iteration
11:   $S_{min} \leftarrow S(\theta^{i+1})$ ;
12:   $\theta^S \leftarrow \theta^{i+1}$ ;
13: end while
14:  $\theta^S \leftarrow$  Parameter vector with important parameters;
15: return  $\theta^S$ .
```

---

We assume that there is an  $\epsilon$ , so that

$$S(\theta^i) = \{\theta^i : \|\theta^i - \theta^S\| < \epsilon\}, \quad (2)$$

$$h_{lm} = \frac{\partial^2 S(\theta^i)}{\partial \theta^l \partial \theta^m} (i \leq l, m \leq d), \quad (3)$$

$$\theta^{i+1} = \theta^i - H^{-1}(\theta^i)g(\theta^i), \quad (4)$$

where  $H^{-1}(\theta^i)$  is the inverse second derivative matrix about  $S$  at the point of  $\theta^i$ ,  $g(\theta^i)$  is the first derivative of  $S(\theta^i)$ ,  $h_{lm}$  belongs to matrix  $H(\theta^i)$ , and  $H^{-1}(\theta^i)g(\theta^i)$  distinguishes the nodes that point to local minimum values in iterations from the nodes point to local non-minimum values.

### 3.3 Multi-layer feed-forward (MLF) neural network

MLF neural network consists of neurons, that are ordered into layers which including input layer, a number of hidden layers and output layer. Fig. 2 and Algorithm 2 both explain how MLF neural network works.

Firstly, the training set  $X_{Train} = \{X_{Train_t}\}$  feeds back to the input layer, where  $t \in [1, p]$ . Secondly, input layer connects hidden layers with weights in  $w_{ij}$ , and hidden layers connect output layer with weights in  $w_{j+n,k}$ . Overall, all neurons are described by a mapping function  $\Gamma$ . Each neuron  $i$  follows the subset  $\Gamma(i) \subseteq V$ , which consists of all ancestors of the given neuron  $i$ . The subset  $\Gamma^{-1}(i) \subseteq V$  then consists of all predecessors of the given neuron  $i$ . The output value of the  $i$ th neuron  $x_i$  is determined by

$$\begin{cases} x_i = f(\xi_i) \\ \xi_i = \vartheta_i + \sum_{j \in \Gamma^{-1}(i)} w_{ij}x_j \\ f(\xi_i) = \frac{1}{1 + e^{-\xi_i}} \end{cases}, \quad (5)$$

where  $\xi_i$  stands for potential of the  $i$ th neuron and function  $f(\xi_i)$  is regarded as a transfer function. If  $\xi_j = 1$ , it means the neuron meets the threshold coefficient, which can be understood as a weight coefficient of the connection with formally added neuron  $j$ . In MLF neural networks, the adaptation process varies  $\vartheta_i$  and  $w_{ij}$  to minimise the variance between the computed and required output values, which evaluated by function  $E$ .

$$E = \sum_o \frac{1}{2} (x_o - \hat{x}_o)^2, \quad (6)$$

where  $x_o$  and  $\hat{x}_o$  are vectors composed of computed output values and required output values, and  $E$  summarises all output neurons  $o$ .

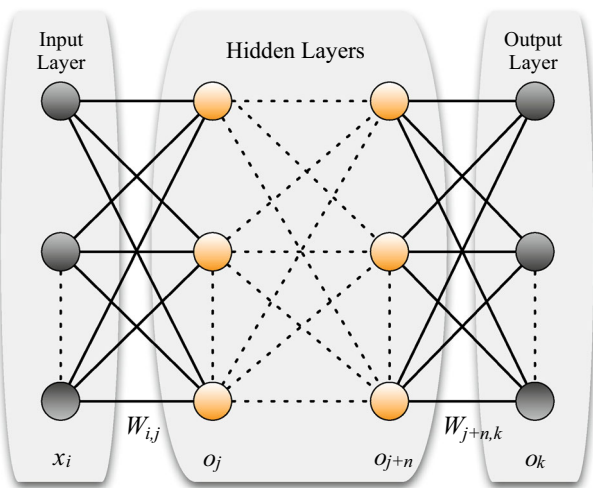


Fig. 2 Structure of MLF neural network

#### Algorithm 2 Multi-layer feed-forward (MLF) neural network

**Input:**  
 Optimized parameter vector  $\theta^S$  output from **Algorithm 1**;  
 Training sets  $X_{Train} = \{X_{Train_t}\}$  under  $\theta^S$  where  $t \in [1, p]$ .

**Output:**  
 Trained  $w_{ij}^{(*)}$  and  $\vartheta_i^{(*)}$ .

// **Initializations**  
 1:  $w_{ij}^{(0)} \leftarrow$  Iteration step 0 on weights of NN layers;  
 2:  $\vartheta_i^{(0)} \leftarrow$  Iteration step 0 on parameter of MLF neural network.  
 3: **while**  $t \leq p$  **do**  
 4:  $\xi_i \leftarrow$  The potentials of the  $i$ th neuron feeds back to input layer, and are calculated via Eq. (4);  
 5:  $f(\xi_i) \leftarrow$  The transfer function calculated via Eq. (4);  
 6:  $x_i \leftarrow$  The  $i$ th neuron feeds back to input layer, calculated on  $f(\xi_i)$  via Eq. (4);  
 // **BP training algorithm for hidden layers**  
 7: **while**  $g_k = 0$  **do**  
 8:  $(\frac{\partial E}{\partial w_{ij}})^{(k)} \leftarrow$  The  $k$ th iteration of  $w_{ij}$  via Eq. (16);  
 9:  $(\frac{\partial E}{\partial \vartheta_i})^{(k)} \leftarrow$  The  $k$ th iteration of  $\vartheta_i$  via Eq. (17);  
 10:  $\lambda^{(k)} \leftarrow$  The learning rate calculated via Eqs. (8)–(14);  
 11:  $w_{ij}^{(k+1)}$  and  $\vartheta_i^{(k+1)} \leftarrow$  Iteration step  $k + 1$  via Eq. (7);  
 12: **end while**  
 13:  $E \leftarrow$  The evaluation function which summaries all output neurons  $o$  via Eq. (15);  
 14:  $\frac{\partial E}{\partial x_i} \leftarrow$  via Eq. (19) or Eq. (20);  
 15:  $w_{ij}^{(o)}$  and  $\vartheta_i^{(o)} \leftarrow$  The trained weight and parameter of MLF neural network;  
 16: **end while**  
 17:  $w_{ij}^{(*)} \leftarrow w_{ij}^{(o)}$ ;  
 18:  $\vartheta_i^{(*)} \leftarrow \vartheta_i^{(o)}$ ;  
 19: **return**  $w_{ij}^{(*)}$  and  $\vartheta_i^{(*)}$ .

In this MLF neural network, we adopt Back-propagation (BP) training algorithm and use Newton method to get minimizations in steepest-descent minimization method.

$$\begin{cases} w_{ij}^{(k+1)} = w_{ij}^{(k)} - \lambda^{(k)} \left( \frac{\partial E}{\partial w_{ij}} \right)^{(k)} \\ \vartheta_i^{(k+1)} = \vartheta_i^{(k)} - \lambda^{(k)} \left( \frac{\partial E}{\partial \vartheta_i} \right)^{(k)} \end{cases}, \quad (7)$$

where  $\lambda > 0$  is the rate of learning.

The  $\lambda$  is a small number that forces the algorithm to make small jumps. In this paper,  $\lambda^{(k)}$  is determined by quasi-Newton method (Loke and Barker, 1996) which is proved to be the best gradient descent approach even for large dimensional problems.

$$\lambda^{(k)} = \min \{ \lambda_{w_{ij}}^{(k)}, \lambda_{\vartheta_i}^{(k)} \} \quad (8)$$

For  $w_{ij}^{(k+1)} = w_{ij}^{(k)} - \lambda_{w_{ij}}^{(k)} \left( \frac{\partial E}{\partial w_{ij}} \right)^{(k)}$ , we have Eqs. (9)–(11).

$$\Delta w_{ij}^{(k)} = w_{ij}^{(k)} - w_{ij}^{(k-1)} \quad (9)$$

$$\Delta \widehat{g}(w_{ij}^{(k)}) = \left( \frac{\partial E}{\partial w_{ij}} \right)^{(k)} - \left( \frac{\partial E}{\partial w_{ij}} \right)^{(k-1)} \tag{10}$$

$$\lambda_{w_{ij}}^{(k)} = \frac{\Delta \widehat{g}(w_{ij}^{(k)})^T \Delta w_{ij}^{(k)}}{\Delta \widehat{g}(w_{ij}^{(k)})^T \Delta \widehat{g}(w_{ij}^{(k)})} \tag{11}$$

For  $\vartheta_i^{(k+1)} = \vartheta_i^{(k)} - \lambda_{\vartheta_i}^{(k)} \left( \frac{\partial E}{\partial \vartheta_i} \right)^{(k)}$ , we have Eqs. (12)–(14).

$$\Delta \vartheta_i^{(k)} = \vartheta_i^{(k)} - \vartheta_i^{(k-1)} \tag{12}$$

$$\Delta \widehat{g}(\vartheta_i^{(k)}) = \left( \frac{\partial E}{\partial \vartheta_i} \right)^{(k)} - \left( \frac{\partial E}{\partial \vartheta_i} \right)^{(k-1)} \tag{13}$$

$$\lambda_{\vartheta_i}^{(k)} = \frac{\Delta \widehat{g}(\vartheta_i^{(k)})^T \Delta \vartheta_i^{(k)}}{\Delta \widehat{g}(\vartheta_i^{(k)})^T \Delta \widehat{g}(\vartheta_i^{(k)})} \tag{14}$$

The derivatives of  $\frac{\partial E}{\partial w_{ij}}$  and  $\frac{\partial E}{\partial \vartheta_i}$  then go to the next process below.

*First step,*

$$E = \frac{1}{2} (x_o - \hat{x}_o)^2 = \frac{1}{2} \sum_k g_k, \tag{15}$$

where  $g_k = x_k - \hat{x}_k \neq 0$  for  $k \in$  output layer, elsewhere  $g_k = 0$ .

*Second step,*

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_i} \frac{\partial f(\xi_i)}{\partial \xi_i} \frac{\partial \xi_i}{\partial w_{ij}} = \frac{\partial E}{\partial \xi_i} f'(\xi_i) x_j, \tag{16}$$

$$\frac{\partial E}{\partial \vartheta_i} = \frac{\partial E}{\partial x_i} \frac{\partial f(\xi_i)}{\partial \xi_i} \frac{\partial \xi_i}{\partial \vartheta_i} = \frac{\partial E}{\partial \xi_i} f'(\xi_i), \tag{17}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \vartheta_i} x_j. \tag{18}$$

---

**Algorithm 3** Finite multi-view mixture distribution

---

**Input:**

Optimized parameter vector  $\theta^S$  output from **Algorithm 1**;

Testing sets  $\mathbf{X}_{Test} = \{\mathbf{X}_{Test_t}\}$  under parameters from  $\theta^S$  where  $t \in [1, p]$ ;

The distributions contribution of each components  $f_j(y | x; \theta_j)$ .

**Output:**

The finite multi-view mixture distribution  $f(y | x; z; \theta_1, \theta_2, \dots, \theta_C; \pi_1, \pi_2, \dots, \pi_C)$ .

- 1:  $\pi_j \leftarrow$  The weights of each component calculated via Eq. (22);
  - 2:  $\lambda_j \leftarrow$  The components of the finite multi-view mixture distribution via Eq. (24);
  - 3:  $E(y_j | x_i) \leftarrow$  The conditional mean of finite mixture distribution via Eq. (24);
  - 4:  $\frac{\partial E(y_j | x_i)}{\partial x_i} \leftarrow$  The marginal effect of finite multi-view mixture distribution via Eq. (25);
  - 5:  $f(y | x; z; \theta_1, \theta_2, \dots, \theta_C; \pi_1, \pi_2, \dots, \pi_C) \leftarrow$  The C-component finite multi-view mixture distribution via Eq. (21);
  - 6: **return**  $f(y | x; z; \theta_1, \theta_2, \dots, \theta_C; \pi_1, \pi_2, \dots, \pi_C)$ .
- 

*Third step,* if  $i \in$  output layer,

$$\frac{\partial E}{\partial x_i} = g_i, \tag{19}$$

else if  $i \in$  hidden layers,

$$\frac{\partial E}{\partial x_i} = \sum_{l=\Gamma_i} \frac{\partial E}{\partial \vartheta_l} w_{ij}. \tag{20}$$

Based on the above algorithm, the output error propagates from the output layer through the hidden layers to the input layer. The layers of MLF neural network depend on the number of hidden layers. For example, if the number of hidden layers is 3, the unit has 4 outputs and this feed-forward neural network is a neural network in 4 multi-layers. The more complex neural network architecture, the more layers feed-forward neural network has. Meanwhile, more weight parameters (neurons) are needed to participate in adaptive systems with a strong training capacity. The optimal number of hidden layers cannot be decided by certain rules. We should repeat the processes for the optimal NN structure as well as the optimal parameters between NN layers. Both processes work on the same training set. The training results directly affect the accuracy of the self-adaptive effect in this framework. Therefore, the accuracy of the estimated accuracy is an important indicator in this study.

**3.4 Finite multi-view mixture distribution**

In general, the multi-source dataset is one kind of heterogeneous datasets and represents data from different groups. The inherent heterogeneity of the data may reflect different phenomena. Through introducing weights to handle heterogeneous data sources with limited datasets, hybrid distribution model is more suitable than a single distribution model in analysis and with more flexibility and agility in prediction. The detailed processes of finite multi-view mixture distribution are shown in Algorithm 3.

Assuming the density function for a C-component finite multi-view mixture is

$$f(y | x; z; \theta_1, \theta_2, \dots, \theta_C; \pi_1, \pi_2, \dots, \pi_C) = \sum_{j=1}^C \pi_j(z) f_j(y | x; \theta_j), \tag{21}$$

where  $0 < \pi_j < 1$ , and  $\sum_{j=1}^C \pi_j = 1$ .

In Eq. (21),

$$\pi_j = \frac{e^{\gamma_j}}{e^{\gamma_1} + e^{\gamma_2} + \dots + e^{\gamma_{C-1}} + 1}, \tag{22}$$

where  $e$  is the mathematical constant of the natural logarithm.



Overall, the objective function for finite multi-view mixture distribution is

$$\max_{\pi, \theta} \ln L = \sum_{i=1}^N \left( \log \left( \sum_{j=1}^C f_j(y) \theta_j \right) \right). \tag{23}$$

The conditional mean of finite multi-view mixture distribution is

$$E(y_i | x_i) = \sum_{j=1}^C \pi_j \lambda_j, \tag{24}$$

where  $\lambda_j = E_j(y_i | x_i)$ .

The marginal effect of finite multi-view mixture distribution is

$$\frac{\partial E(y_i | x_i)}{\partial x_i} = \sum_{j=1}^C \pi_j \frac{\partial E_j(y_i | x_i)}{\partial x_i} = \sum_{j=1}^C \pi_j \frac{\partial \lambda_j}{\partial x_i}. \tag{25}$$

## 4 Simulation and results

### 4.1 Sample selection and pre-process

The simulation sample is selected from Shanghai Statistical Yearbooks (2001–2014) (Shanghai Statistical Department 2015). Shanghai Statistical Yearbooks (2001–2014) published year-data from 2000 to 2013. We choose 25 attributes from 6 views which are related to information services of cloud ITS. The sample structure is shown in Table 1. In this simulation, we aim to

- Test Newton iteration method used in the data pre-process and the MLF neural network training;
- Test MLF neural network training embedded in multi-view information service system;

- Test finite multi-view mixture distribution fitting for near future predictions; and
- Reflect passengers’ behaviors from different transportation modes and travel payments.

By this simulation, we try to prove that the proposed self-adaptive multi-view framework is practicable for multi-source information service in cloud ITS.

The original samples, which are selected from Shanghai Statistical Yearbooks, lose one or more specific values in some attributes. As the first part of our simulation, we jointly use MATLAB and SPSS Statistics to pre-process the original samples in dimensions of  $14 \times 25$  before training.

Firstly, we use point linear trend method to process the missing values and test each t-significances. The results shown in Table 2 indicated that these 6 missing values in 4 attributes and 2 views are well replaced with a very good significant test results.

Secondly, values from different attributes are processed into an unified dimension by Z-score method (see Eq. (26)) to standardize all the index values with means of 0 and standard deviation of 1, so that attributes with diverse unites can be compared and analysed.

$$\begin{cases} \hat{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{S_j}, i = 1, \dots, 14 \text{ and } j = 1, \dots, 25 \\ \bar{x}_j = \frac{1}{14} \sum_{i=1}^{14} x_{ij}, j = 1, \dots, 25 \\ S_j = \sqrt{\frac{1}{14} \sum_{i=1}^{14} (x_{ij} - \bar{x}_j)^2}, j = 1, \dots, 25 \end{cases}, \tag{26}$$

where  $\hat{x}_{ij}$  is standardized data,  $\bar{x}_j$  is the average of  $j$ , and  $S_j$  is the standard indicator of  $j$ .

Thirdly, we compare the Newton iterative method with the traditional dimensional reduction method of factor

**Table 1** Structure of the simulation sample (2000-2013)

View	Details: items (codes)
Road infrastrucure (View #1)	Length of viaduct (LoV), amount of bridges over Huangpu River (AoHPRB), and amount of tunnels across Huangpu River (AoHPRT)
Rail transit (View #2)	Amount of operating vehicles (AoOV), length of operation lines (LoOL), operating distance (DoOR), and passengers volume (VoPR)
City bus (View #3)	Length of bus lines (LoBL), amount of bus lines (AoBL), amount of operating bus (AoOB), and passengers amount (AoPB)
Taxi (View #4)	Amount of taxis (AoT), amount of taxis with passengers (AoPT), operating distance (DoOT), and amount of operating corporations (AoOCT)
Information service in public transport (View #5)	Registers amount of Public Mailbox (RAoPM), trading amount of payment platform (TAoPP), sales amount of public transport cards (SAoPTC), turnover of payment platform (ToPP), turnover of public transport cards (ToPTC), sales amount of bank cards (SAoBC), and turnover of bank cards (ToBC)
Social credit system (View #6)	Input amount of personal credit information (IAoPCI), reports amount of personal credit (RAoPC), and input amount of corporate credit information (IAoCCI)

**Table 2** Replaced missing value and its t-Sig

Attribute	View	Year	Sig.
SAoBC	View #5	2011, 2012, 2013	0.001
IAoPCI	View #6	2013	0.000
RAoPC	View #6	2013	0.000
IAoCCI	View #6	2013	0.000

(\* Confidence interval of t-test is 95 %)

analysis (Harman 1960). The comparisons are shown in Table 3.

Obviously, the Newton iterative method has a better performance than factor analysis which is a most popular dimensional reduction method. In this simulation, 23 of 25 attributes are extracted by factor analysis which carry 94.546 % information of the selected samples. However, by the Newton iterative method, the attribute dimension is reduced to 12, that is 95.453 percent of information from samples can be represented by the 12 attributes. Moreover, the testing result shows that the extracted attributes can cover most information views (see Table 4). Actually, the only abandoned view (View #3) can be concluded into View #5. Therefore, Newton iterative method is proved to be efficient in ITS data's reductions.

Finally, the data in 12 attributes and 5 views below are input to the MLF neural network training and finite mixture distribution fitting in next two sections.

#### 4.2 Training results of MLF neural network

In this section, we use MATLAB to process the Newton iterative method based MLF neural network training and compare with other NNs, i.e., dynamic NN (Shaw et al.

1997), prune NN (Karnin 1990) and Bayesian multi-layer NN (Auld et al. 2007), jointly by MATLAB and SPSS Statistics. The basis settings for Newton iterative method based MLF neural network training and its comparisons are,

- Selected samples in 25 attributes of 6 views are input to MLF neural network, while data trained by Newton iterative method are used as target data for MLF neural network. The data structure of target data are shown in Table 4.
- We set the the first 70 % samples as the training dataset and the rest 30 % as testing dataset.
- Regarding small sample size, we set 2 hidden layers for each MLF neural network training, i.e., Bayesian multi-layer NN and Newton iterative method based MLF neural network.
- We firstly train Newton iterative method based MLF neural network to get the momentum result in which the MLF neural network obtains optimal weights by BP training. Then we adopt this iteration to prune NN and Bayesian multi-layer NN trainings.
- Dynamic NN, Prune NN and Bayesian NN share the same neuron amounts of input layer and output layer with Newton iterative method based MLF neural network.
- In this simulation, we test learning rate from 0.5 to 1 to obtain best estimated accuracy. MLF neural network with BP training totally relies on the learning rate  $R$ . (see in Fig. 3).

Based on above settings, we obtain the results from 4 different NNs in Table 5. To make a clear explanation, 2 comparisons are going to be presented below.

*Comparison 1. Multi-layer NN vs. Single layer NN* Dynamic NN which has single hidden layer is iterated 35

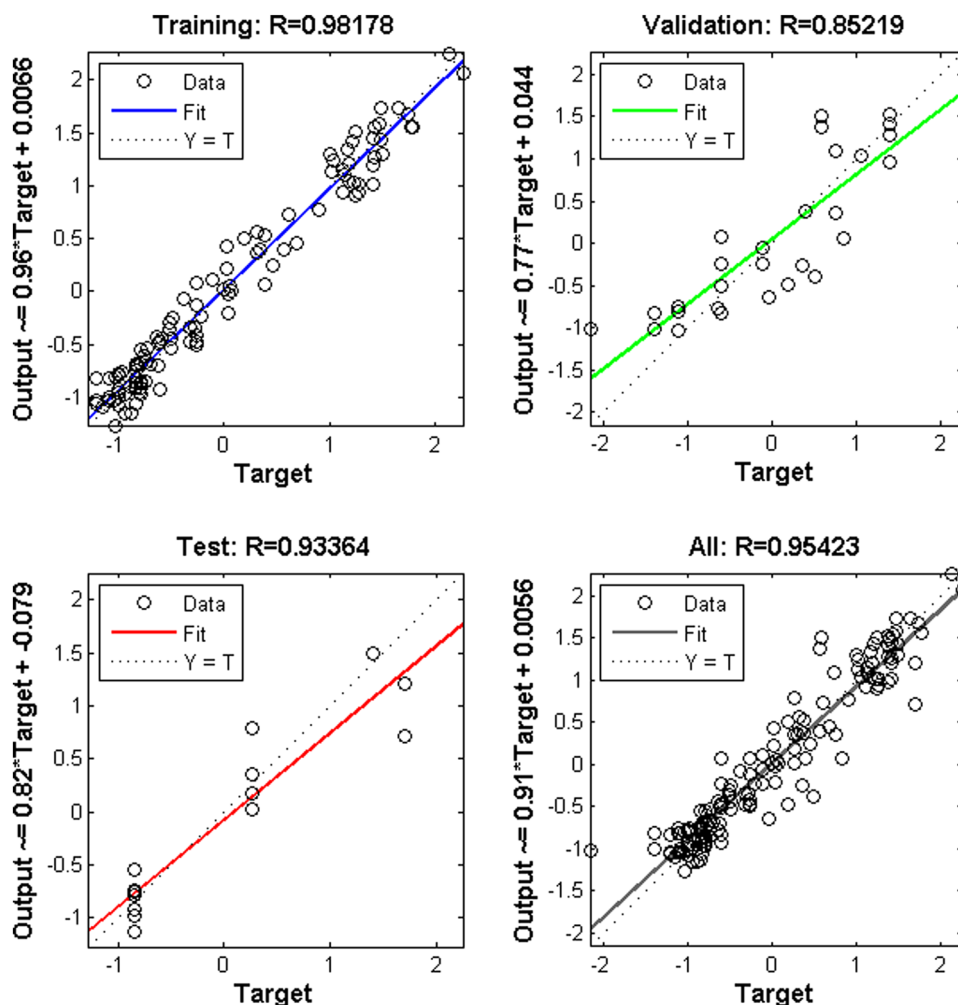
**Table 3** Newton iterative method vs. factor analysis

	Amount of attributes	Amount of views	Cumulative extractions (%)
Selected sample	25	6	100
Factor analysis (Harman 1960)	23	6	94.546
Newton iterative method	12	5	95.453

**Table 4** Data for MLF Neural network training

View	Details: items (codes)
Road infrastructure (View #1)	Length of viaduct (LoV), amount of bridges over Huangpu River (AoHPRB), and amount of tunnels across Huangpu River (AoHPRT)
Rail transit (View #2)	Amount of operating vehicles (AoOV), length of operation lines (LoOL), operating distance (DoOR), and passengers volume (VoPR)
Taxi (View #4)	Amount of taxis with passengers (AoPT)
Information service in public transport (View #5)	Turnover of payment platform (ToPP), turnover sales amount of bank cards (SAoBC), and turnover of bank cards (ToBC)
Social credit system (View #6)	Input amount of corporate credit information (IAoCCI)

**Fig. 3** Learning Rate of MLF Neural Network on Newton Iterative Method



times to obtain accuracy on 93.018 %. Prune NN which also has single hidden layer is iterated 6 times to obtain accuracy on 95.094 %. However, our Newton iterative method based MLF neural network which has 2 hidden layers is iterated 52 times to obtain accuracy on 98.590 % which is higher than the two single layer NN. Besides, red curves in Fig. 4a, b appear above blue and green curves while red curves in Fig. 4c is below blue and green curves, which means trained dynamic NN and prune NN do not work well when new data input. In real-world application, multi-layer NN will have stronger self-adaptation than single layer NN. When new data input into NNs, multi-layer NNs can compute faster than single layer NNs as they re-train fewer parts than single layer NNs.

*Comparison 2. Multi-layer feed-forward NN vs. Multi-layer NN* To make comparison between two different multi-layer NNs, we take Bayesian multi-layer NN into consideration. In Table 5, iterations in Bayesian multi-layer NN is far higher than Newton MLF neural network, which means Newton MLF neural network trains faster than

Bayesian multi-layer NN. Moreover, Newton MLF neural network obtain a good estimated accuracy in such fewer iterations. Other parameters for these two NNs are quite similar.

Therefore, it is feasible to apply the MLF neural network to our self-adaptive multi-view framework for multi-source information services in cloud ITS.

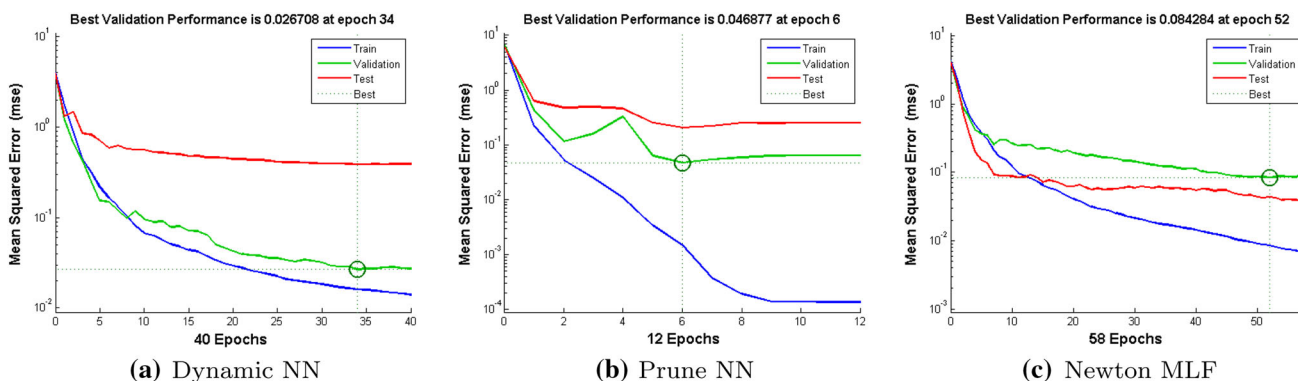
### 4.3 Fitting result of finite multi-view mixture distribution

We used MATLAB and SPSS Statistical to fit the finite mixture distribution. For comparisons, we fit the distributions for each single attributes as well (see Fig. 5). The fitting results is shown in Table 6 where X-axis is the time axis. The training data is set in 70 and 30 % for testing as well.

By training above  $f_i(y|x)$ , we obtain optimal  $\{\theta_1, \dots, \theta_{12}\}$  and  $\{\pi_1, \dots, \pi_{12}\}$  for finite multi-view mixture distribution, where domain  $\{\theta_1, \dots, \theta_{12}\}$  corresponds

**Table 5** Newton iterative method based MLF neural network vs. Other NNs

Method of NNs	Estimated accuracy (%)	Iterations	Mu.	Learning rate	Amount of neurons			
					Input layer	Hidden layers		Output layer
						No.1	No.2	
Dynamic (Shaw et al. 1997)	93.018	35	/	/	25	17	/	12
Prune (Karnin 1990)	95.094	6	5	/	25	17	/	12
Bayesian ML (Auld et al. 2007)	98.015	704	5	0.946	25	12	10	12
Newton MLF	98.590	52	5	0.954	25	12	10	12



**Fig. 4** Comparison 1: Performances of dynamic NN, prune NN and Newton iterative method based MLF neural network

to views and its importance in each view from view-local results of Newton iterative method. The finite multi-view mixture distribution is,

$$f(y|x) = -1.277 + 0.06x + 0.014x^2 + 0.00025x^3. \quad (27)$$

The finite multi-view mixture distribution is represented in Fig. 6 on the blue bold curve together with curves in 5 views. In Fig. 6, the noise of multi-view mixture distribution fitting comes from view #4 taxi. Obviously, distribution of taxi service is different from other public transportations. However, our finite multi-view mixture distribution succeeds in avoid noises.

**4.4 Discussion**

Through the simulation processes, which consist of sample selection and pre-process by Z-score and Newton iterative method, MLF neural network training on Newton iterative method, and finite multi-view mixture distribution fitting on Newton iterative method, the self-adaptive multi-view framework for multi-source information service in cloud ITS is proved to be feasible.

- The Newton iterative method for multi-parameter optimization goes well with the MLF neural network.

- The MLF neural network can fully train disorganized multi-source samples in a less supervised non-linear way, which is suitable for our self-adaptive multi-view framework. Meanwhile MLF neural network can always offer general linkings to complex datasets in real world.
- The finite mixture algorithm flexibly fits multiple distribution in a common time line. The finite multi-view mixture distribution clearly shows present patterns and future trends. The finite multi-view mixture distribution also significantly avoid noise.

**5 Conclusion**

Cloud computing, a socialized and specialized tool for data-driven information services especial in a big data requirement, is becoming a new network service mode. Because of cloud computing, fusion mechanism of multi-view information and its management methods are changing, which provide a powerful impetus for innovations of multi-view information service framework and its integrations of multiple sources. Therefore, through integrations of multi-view information, providing accurate

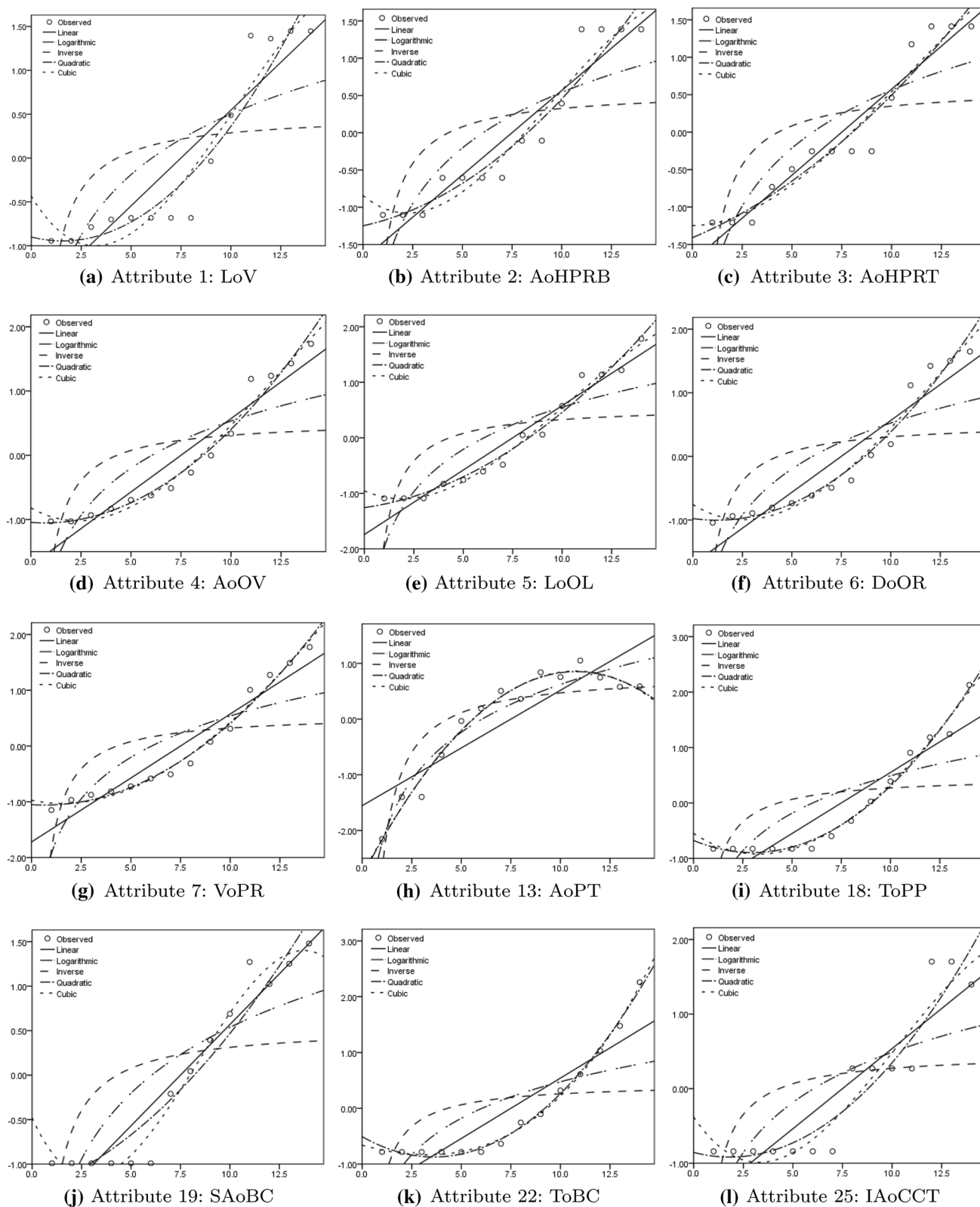
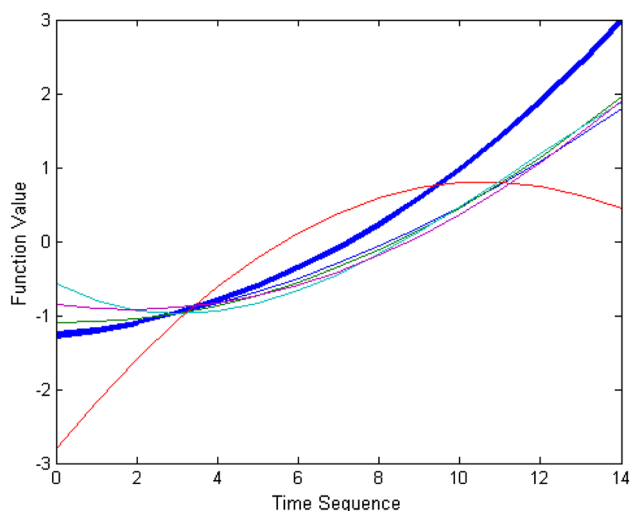


Fig. 5 Fitting curves from 12 single attributions

**Table 6** Fitting results of each single attributes

Attr.	Function
1	$y = -0.901 - 0.058x + 0.018x^2$
2	$y = -1.248 + 0.057x + 0.011x^2$
3	$y = -1.410 + 0.112x + 0.008x^2$
4	$y = -1.045 - 0.022x + 0.017x^2$
5	$y = -1.257 + 0.050x + 0.012x^2$
6	$y = -0.977 - 0.043x + 0.018x^2$
7	$y = -1.051 - 0.022x + 0.017x^2$
13	$y = -2.817 + 0.681x - 0.032x^2$
18	$y = -0.676 - 0.146x + 0.024x^2$
19	$y = -0.481 - 0.487x + 0.101x^2 - 0.004x^3$
22	$y = -0.509 - 0.201x + 0.028x^2$
25	$y = -0.855 - 0.069x + 0.019x^2$

**Fig. 6** Curves of finite multi-view mixture distribution on global and each view

information to users will be the key information services in ITS which is a so-called data-driven ITS.

In this paper, we propose a self-adaptive multi-view framework for multi-source information service in cloud ITS. We expect this framework to study various behaviours of cloud ITS users, which included end-uses, services providers, etc., to output user-oriented information services, and to improve traffic management in mega-cities such as Shanghai. This study also proves the self-adaptive functions of this framework to be feasible by simulations.

In current study, we have not considered applying multi-view information into a data-driven decision support system especially for cloud ITS. By applying cloud computing techniques with big data network structures to multi-view decision support system can be a good way to solve multi-view information services problems. As future work, the

decision support system of self-adaptive multi-view information service in cloud ITS is planned, which can output multi-view service information for cloud ITS users and help users make personal traffic plans.

**Acknowledgments** The project is funded by Shanghai science and technology committee under Grant No. 14692105900, by Shanghai education commission under Grant No. 14ZS085, and by education ministry of China under Grant No. 12YJA630158.

## References

- Ali K, Al-Yaseen D, Ejaz A, Javed T, Hassanein HS (2012) Crowdsits: Crowdsourcing in intelligent transportation systems. In: Wireless Communications and Networking Conference (WCNC), 2012 IEEE, IEEE, pp 3307–3311
- Ashokkumar K, Sam B, Arshadprabhu R (2015) Cloud based intelligent transport system. *Procedia Comput Sci* 50:58–63
- Auld T, Moore AW, Gull SF (2007) Bayesian neural networks for internet traffic classification. *IEEE Trans Neural Netw* 18(1):223–239
- Balcazar J, Gavalda R, Siegelmann H (1997) Computational power of neural networks: a characterization in terms of kolmogorov complexity. *IEEE Trans Inf Theory* 43(4):1175–1183
- Barceló J, Codina E, Casas J, Ferrer JL, Garcia D (2005) Microscopic traffic simulation: a tool for the design, analysis and evaluation of intelligent transport systems. *J Intell Robot Syst* 41(2–3):173–203
- Biem A, Bouillet E, Feng H, Ranganathan A, Riabov A, Verscheure O, Koutsopoulos H, Moran C (2010) Ibm infosphere streams for scalable, real-time, intelligent transportation services. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, pp 1093–1104
- Branch R, Tjeerdsma H, Wilson C, Hurley R, McConnell S (2014) Cloud computing and big data: a review of current service models and hardware perspectives. *J Softw Eng Appl* 7(8):686–693
- Chen C, Zhang DQ, Li N, Zhou ZH (2014) B-planner: Planning bidirectional night bus routes using large-scale taxi gps traces. *IEEE Trans Intell Transp Syst* 15(4):1451–1465
- Cireşan D, Meier U, Masci J, Schmidhuber J (2012) Multi-column deep neural network for traffic sign classification. *Neural Netw* 32:333–338
- Colombaroni C, Fusco G (2014) Artificial neural network models for car following: experimental analysis and calibration issues. *J Intell Trans Syst* 18(1):5–16
- Delot T, Ilarri S, Thilliez M, Vargas-Solar G, Lecomte S (2011) Multi-scale query processing in vehicular networks. *J Ambient Intell Humaniz Comput* 2(3):213–226
- Dikaiakos MD, Katsaros D, Mehra P, Pallis G, Vakali A (2009) Cloud computing: distributed internet computing for it and scientific research. *IEEE Internet Comput* 13(5):10–13
- Dinakaran M (2014) Intelligent transportation systems. *Indian. Highways* 42(5):57–65
- Duan Q, Yan YH, Vasilakos AV (2012) A survey on service-oriented network virtualization toward convergence of networking and cloud computing. *IEEE Trans Netw Serv Manag* 9(4):373–392
- El Faouzi NE, Leung H, Kurian A (2011) Data fusion in intelligent transportation systems: progress and challenges—a survey. *Inf Fusion* 12(1):4–10
- Hao L, Lei X, Yan Z, ChunLi Y (2012) The application and implementation research of smart city in china. In: System

- Science and Engineering (ICSSE), 2012 International Conference on, IEEE, pp 288–292
- Harman HH (1960) Modern factor analysis. University of Chicago Press, Chicago
- Hoffmann P, Feng SC, Ameta G, Ghodous P, Qiao LH (2008) Towards a multi-view semantic model for product feature description. In: Collaborative Product and Service Life Cycle Management for a Sustainable World, Springer, pp 205–213
- Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, IEEE, vol 2, pp 985–990
- Huang SC, Jiau MK, Lin CH (2015) A genetic-algorithm-based approach to solve carpool service problems in cloud computing. *IEEE Transactions Intell Transp Syst* 16(1):352–364
- Jain AK, Mao J, Mohiuddin K (1996) Artificial neural networks: a tutorial. *Computer* 29(3):31–44
- Karnin ED (1990) A simple procedure for pruning back-propagation trained neural networks. *IEEE Trans Neural Netw* 1(2):239–242
- Koehler LA, Kraus W Jr, Camponogara E (2011) Iterative quadratic optimization for the bus holding control problem. *IEEE Trans Intell Transp Syst* 12(4):1568–1575
- Kramers A (2014) Designing next generation multimodal traveler information systems to support sustainability-oriented decisions. *Environ Model Softw* 56:83–93
- Leu FY, Lin CH, Castiglione A (2013) Special issue on cloud, wireless and e-commerce security. *J Ambient Intell Humaniz Comput* 4(2):207–208
- Leviäkangas P, Haajanen J, Alarukka AM (2007) Information service architecture for international multimodal logistic corridor. *IEEE Trans Intell Transp Syst* 8(4):565–574
- Li XF, Xu JP, Wang YQ, He CZ (2004) The establishment of self-adapting algorithm of bp neural network and its application. *Syst Eng Theory Pract* 5:1
- Li ZJ, Chen C, Wang K (2011) Cloud computing for agent-based urban transportation systems. *IEEE Intell Syst* 26(1):73–79
- Lim MK, Tang S, Chan CS (2014) Isurveillance: Intelligent framework for multiple events detection in surveillance videos. *Expert Syst Appl* 41(10):4704–4715
- Liu F, Tong J, Mao J, Bohn R, Messina J, Badger L, Leaf D (2011) Nist cloud computing reference architecture. *NIST Spec Publ* 500:292
- Liu JY, Wang WD, Gong XY, Que XR, Yang H (2012) A hybrid model based on kalman filter and neural network for traffic prediction. In: Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on, IEEE, vol 2, pp 533–536
- Loke MH, Barker R (1996) Rapid least-squares inversion of apparent resistivity pseudosections by a quasi-newton method. *Geophys Prospect* 44(1):131–152
- Ma YC, Chowdhury M, Sadek A, Jekhiani M (2012) Integrated traffic and communication performance evaluation of an intelligent vehicle infrastructure integration (vii) system for online travel-time prediction. *IEEE Trans Intell Transp Syst* 13(3):1369–1382
- Maleki-Dizaji S, Siddiqi J, Soltan-Zadeh Y, Rahman F (2014) Adaptive information retrieval system via modelling user behaviour. *J Ambient Intell Humaniz Comput* 5(1):105–110
- Matthew DS (2012) Multi-parameter optimization: identifying high quality compounds with a balance of properties. *Current Pharm Des* 18(9):1292
- Mell P, Grance T (2011) The nist definition of cloud computing. Tech. rep, Gaithersburg
- Merzouki R, Samantaray AK, Pathak PM, Bouamama BO (2013) Intelligent transportation systems. In: Intelligent Mechatronic Systems, Springer, pp 769–867
- Mugellini E, Sokhn M, Szczepaniak PS (2013) Special issue on intelligent techniques for multimedia information management and personalization. *J Ambient Intell Humaniz Comput* 4(3):347–348
- Ramesh R, Durbar YS, Jayanth J (2013) Intelligent transport system using cloud platform. In: Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on, IEEE, pp 1071–1075
- Rigatos G, Siano P, Zervos N (2013) An approach to fault diagnosis of nonlinear systems using neural networks with invariance to fourier transform. *J Ambient Intell Humaniz Comput* 4(6):621–639
- Sanger TD (1989) Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Netw* 2(6):459–473
- Saukh O, Hasenfratz D, Thiele L (2014) Route selection for mobile sensor nodes on public transport networks. *J Ambient Intell Humaniz Comput* 5(3):307–321
- Shanghai Government (2011) Shanghai's smart city strategy (2011–2013). <http://www.shanghai.gov.cn/shanghai/node2314/node2-319/node12344/u26ai28792.html>. Accessed 5 June 2015
- Shanghai Statistical Department (2015) Shanghai Statistical Yearbooks, vol 2001–2014. <http://www.stats-sh.gov.cn/data/release.xhtml>. Accessed 5 June 2015
- Shaw AM, Doyle FJ, Schwaber JS (1997) A dynamic neural network approach to nonlinear process modeling. *Comput Chem Eng* 21(4):371–385
- Sherman AH (1978) On newton-iterative methods for the solution of systems of nonlinear equations. *SIAM J Numer Anal* 15(4):755–771
- Stilwell DJ, Bay JS (1993) Toward the development of a material transport system using swarms of ant-like robots. In: Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on, IEEE, pp 766–771
- Su HS (2008) Multi-source fuzzy information fusion method based on bayesian optimal classifier. *Acta Autom Sin* 34(3):282–287
- Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. *J Netw Comput Appl* 34(1):1–11
- Tao CC (2007) Dynamic taxi-sharing service using intelligent transportation system technologies. In: Wireless Communications, Networking and Mobile Computing, International Conference on, IEEE, pp 3209–3212
- Trivedi P, Deshmukh K, Shrivastava M (2012) Cloud computing for intelligent transportation system. *Int J Soft Comput Eng* 2(3):568–572
- Wan JF, Zhang DQ, Sun YT, Lin K, Zou CF, Cai H (2014) Vcmia: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing. *Mobile Netw Appl* 19(2):153–160
- Wang FY (2010) Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications. *IEEE Trans Intell Transp Syst* 11(3):630–638
- Wang L, Wang CJ, Shen XR, Fan YZ (2005) A new method of real-time information collection in intelligent transportation system. *Syst Eng* 23(2):86–89
- Wu J, Cai ZH (2014) A naive bayes probability estimation model based on self-adaptive differential evolution. *J Intell Inf Syst* 42(3):671–694
- Wu J, Cai ZH, Zeng SY, Zhu XQ (2013a) Artificial immune system for attribute weighted naive bayes classification. In: Neural Networks (IJCNN), The 2013 International Joint Conference on, pp 798–805
- Wu J, Cai ZH, Zhu XQ (2013b) Self-adaptive probability estimation for naive bayes classification. In: Neural Networks (IJCNN), The 2013 International Joint Conference on, pp 2303–2310

- Wu J, Zhu XQ, Zhang CQ, Yu PS (2014) Bag constrained structure pattern mining for multi-graph classification. *IEEE Trans Knowl Data Eng* 26(10):2382–2396
- Wu J, Pan SR, Zhu XQ, Cai ZH (2015a) Boosting for multi-graph classification. *IEEE Trans Cybern* 45(3):430–443
- Wu J, Pan SR, Zhu XQ, Cai ZH, Zhang P, Zhang CQ (2015b) Self-adaptive attribute weighting for naive bayes classification. *Expert Syst Appl* 42(3):1487–1502
- Xie MQ, Li XM, Zhou WL, Fu YB (2014) Forecasting the short-term passenger flow on high-speed railway with neural networks. *Comput Intell Neurosci* 2014:23
- Yager RR (2005) Some considerations in multi-source data fusion. In: *Intelligent Data Mining*, Springer, pp 1–22
- Yang XY, Nasser B, Surrige M, Middleton S (2012) A business-oriented cloud federation model for real-time applications. *Future Gener Computer Syst* 28(8):1158–1167
- Zhang C, Yang Y, Du Z, Ma C (2015) Particle swarm optimization algorithm based on ontology model to support cloud computing applications. *J Ambient Intell Humaniz Comput* 1–6
- Zhang J, Wang FY, Wang K, Lin WH, Xu X, Chen C (2011) Data-driven intelligent transportation systems: a survey. *IEEE Trans Intell Transp Syst* 12(4):1624–1639
- Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. *J Internet Serv Appl* 1(1):7–18
- Zhou YL (2014) The path towards smart cities in china: From the case of shanghai expo 2010. In: *Proceedings Real Corp*, pp 1023–1027