

Article

Urban Traffic Congestion Evaluation Based on Kernel the Semi-Supervised Extreme Learning Machine

Qing Shen *, Xiaojuan Ban * and Chong Guo

University of Science and Technology Beijing, Beijing 100083, China; kotrue2015@gmail.com

* Correspondence: shenqingcc222333@gmail.com (Q.S.); banxj@ustb.edu.cn (X.B.);

Tel.: +86-138-1036-3794 (Q.S.); +86-136-8310-8947 (X.B.)

Academic Editor: Yuhua Luo

Received: 20 January 2017; Accepted: 4 May 2017; Published: 10 May 2017

Abstract: There is always an asymmetric phenomenon between traffic data quantity and unit information content. Labeled data is more effective but scarce, while unlabeled data is large but weaker in sample information. In an urban transportation assessment system, semi-supervised extreme learning machine (SSELM) can unite manual observed data and extensively collected data cooperatively to build connection between congestion condition and road information. In our method, semi-supervised learning can integrate both small-scale labeled data and large-scale unlabeled data, so that they can play their respective advantages, while the ELM can process large scale data at high speed. Optimized by kernel function, Kernel-SSELM can achieve higher classification accuracy and robustness than original SSELM. Both the experiment and the real-time application show that the evaluation system can precisely reflect the traffic condition.

Keywords: traffic congestion evaluation; Kernel-SSELM; asymmetric data; cooperative learning

1. Introduction

Urban road networks are an integral part of the organic link to achieve mutual coordination for a city. Road traffic congestion is an important indicator to measure the level of road service and traffic capacity. Based on the collection of the floating car data combined with the urban road conditions to achieve urban congestion assessment is the main method of the current study.

In most previous researches, the value of congestion is often obtained in two ways. One is to generate assessment models directly by analyzing a large number of unlabeled traffic data [1–4]. The alternative approach is judged by the citizens or traffic governors [5,6]. They make a subjective assessment of congestion and produce some labeled samples, based on the observation of road conditions. Then the evaluation model is built by learning the experience of those labeled samples.

However, there is always an asymmetric phenomenon between traffic data quantity and unit information content. Unlabeled data can be easily obtained so that the number is huge, but the deficiency of subjective evaluation information means evaluation results based on unlabeled data usually deviate to match the public's feeling. Labeled data reflect the public's direct experience and has more information, but the number is fewer because the labeled samples are difficult to obtain. This leads to a gap between the model generated by the first method and the users' experience, and the poor generalization performance of the second methods.

In this paper, kernel based semi-supervised extreme learning machine is used to train the traffic congestion evaluation framework, with both small-scale labeled data and large-scale unlabeled data. In our method, semi-supervised learning can integrate two kinds of data, and use their respective advantages, while the Extreme Learning Machine (ELM) can process large scale data at high speed and the kernel function can increase the stability of the model. Based on this model, we build an evaluation

system to clearly show the traffic congestion on the web map, which can provide assistance for traffic control and public travel.

In our work, the Urban Transportation Assessment and Forecast System analyzes the traffic congestion of the transportation network in a city of southwest China and shows the evaluation results of the real-time traffic states on the Geographic Information System (GIS) map using different colors (Figure 1).

The traffic congestion evaluation system based on floating car data is the fundamental part of core function whose data source consists of road sections information (containing road grades, the number of lanes, the number of neighborhood lanes) and floating car data. We pre-process the floating car data and match the effective floating car speed information to every road section. Finally, road section traffic flow eigenvalues are calculated.

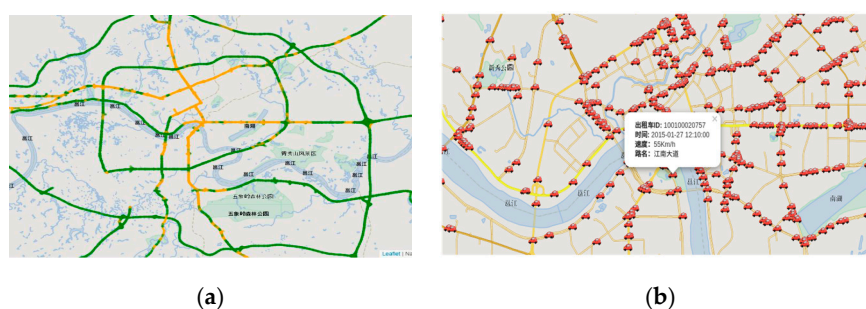


Figure 1. The user interface of the traffic congestion evaluation system: (a) Congestion evaluation on the map; (b) Floating car distribution on the map.

In previous work on traffic congestion evaluation, the empirical evaluation frameworks [1–3] are adopted to build a connection between average speed of floating cars and congestion. These frameworks were based on the analysis of large amounts of unlabeled data. Table 1 shows the road congestion evaluation framework of Beijing. This method is easy to implement and consumes few system resources.

Table 1. Empirical framework based on average speed (km/h).

Congestion	Smooth	Average	Congested
Highway	>65	35~65	<35
Main road	>40	30~40	<20
Minor road and Branch road	>35	25~35	<10

However, it is only possible to get the time distributions and spatial distributions of the traffic flow based on floating car data while the traffic density and flow are unpredictable. Therefore, the empirical evaluation frameworks do not take full consideration of the road information and network conditions. The fixed evaluation standards lack flexibility and do not truly reflect specific urban traffic conditions, traffic regulations and residents' actual feelings about congestion, which results in the gap between the congestion information on the map and users' experience.

To overcome the shortcomings above, intelligent methods are introduced into the traffic congestion evaluation system. Aimed at specific urban traffic conditions, more precise evaluation models are built by least absolute shrinkage and selection operator (LASSO) [4], Neural Network [5], hidden markov model (HMM) [6], support vector machine (SVM) [7], etc. However, these approaches have the following deficiencies when applied to the traffic congestion evaluation system based on floating car data.

- It is difficult to attain the reliable labeled data. Data resources are more than 23,000 taxicabs with GPS in the city. The quantity of data from the floating cars each week is more than

15,000,000 samples and the quantity of road traffic samples is around 700,000. The reliable labeled data of congestion is derived from the real-time observations of the Transportation Department staff that cost much human resources and working time. The traffic network of the city is complex on account of different kinds of bridges, tunnels, main roads, sub roads and intersections. The evaluation models based upon the supervised learning are ineffective because of the sparsely-labeled samples.

- For many semi-supervised learning algorithms, large scale data results in huge computation cost. With the continuous change of traffic conditions, the evaluation model needs to be retrained frequently. So this application demands a machine learning framework that offers more stable and efficient training.

ELM was designed for single hidden layer feedforward neural networks with unique advantages such as high training efficiency, easy implementation and unification of multi-classification and regression [8]. Recently, ELM has gained great progresses in different research field. Incremental ELM [9] makes hidden nodes incrementally constructed and Sequential ELM [10] makes input data processed sequentially. The multilayer learning architecture that uses ELM auto-encoder [11] and subnetwork nodes [12] expands ELM from a single layer structure to a multilayer structure. Also, recent applications of ELM have included: machine vision [13,14], ensemble learning [15,16], sparse learning [17,18] big data applications [19,20], etc.

Semi-supervised ELM [21] extends the capacity to deal with unlabeled data with high training efficiency and accuracy. Kernel based methods [22,23] solved the problem resulting from random distribution of hidden layer parameters in ELM and gain higher relevance to corresponding datasets as well as higher stability. Optimized by kernel function, semi-supervised extreme learning machine (SSELM) can achieve higher classification accuracy and robustness. This paper applied Kernel-SSELM in a traffic congestion evaluation system to unite small-scale labeled data and large-scale unlabeled data cooperatively, and build a connection between congestion conditions and road information with high efficiency and accuracy.

Applying Kernel-SSELM to a traffic congestion evaluation system based on floating car data, this paper has the following strengths:

- Though the congestion value of unlabeled data is uncertain, it represents the different traffic conditions which reflect the distribution information of traffic data. Kernel-SSELM improves the recognition accuracy of evaluation models by involving unlabeled data in the training.
- Extreme learning machine has high training efficiency and is easy to implement. In the case of large data scales, high training speed ensures that the congestion evaluation model can be updated in time according to the data changes.
- In neglecting the number of hidden layer nodes, the optimization of kernel function improves the stability of SSELM.

Our previous work [24] introduced Kernel-SSELM to traffic evaluation. In this paper, we introduce the feature extraction of the congestion (Section 2) and the training algorithm of the evaluation model (Section 3) in detail. Furthermore, based on the training experiment for feasibility verification (Section 4), we have implemented the multi-level evaluation system (Section 5) from the road level to the zone level.

2. Traffic Congestion Eigenvalue

Traffic congestion evaluation system takes the road sections as the individual samples. Specifically, a road section demonstrates a portion of a road in a single direction. The traffic congestion evaluation of each road section originates from two sources cooperatively.

2.1. Road Section Information

The first source is the essential information of the road section from the Transportation Department. The following information is extracted:

- Number of lanes:

The number of lanes represents the road capacity. Under the circumstances of same congestion, the more lanes there is, higher the speed is.

- Number of Entrance and Exit:

These two numbers represent the capacity of the road's input and output. This capacity has an obvious effect on stopping time and speed of cars in the different conditions of congestion.

- Number of traffic lights:

The number of traffic lights influences the stopping time and average speed of cars on the road, especially at an intersection. Despite the long stopping and low average speed, if cars have a certain velocity and there are some traffic lights, the road will still remain smooth.

- Road grades:

Roads in the system have four road grades. They are: highway, major road, minor road and branch way. This eigenvalue is defined with the numerical value. From branch way to highway the numerical values are 1, 2, 3, 4, respectively. The higher the road grade is, the higher the road standard speed and maximum speed the road has. The congestion evaluation is also different because of different road grades.

2.2. Speed Information Based on Floating Car Data

The second source is the real-time speed information of the road section from the floating car data. In the interval ΔT defined by $\Delta T = 5$ min, the data is calculated and matched to the corresponding road section and is transformed to the following kinds of eigenvalues:

- Average speed:

First, we calculate the average speed of every car in the interval of ΔT on the road section. Then the average speed of all cars is attained. In the interval of ΔT , the velocity measurement sites of the floating car r on the terminal road section are distributed as shown in the Figure 2.

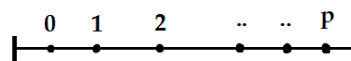


Figure 2. The distribution of the move trajectory of a floating car.

Sequence $\{t_0, t_1, \dots, t_p\}$ and sequence $\{u_0, u_1, \dots, u_p\}$ are the time sequence and speed sequence of floating car r on the road section. The floating car's driving distance S_r is defined by:

$$S_r = \int_{t_0}^{t_p} u dt \approx u_0 \left(\frac{t_1 - t_0}{2} \right) + u_p \left(\frac{t_p - t_{p-1}}{2} \right) + \sum_{i=1}^{p-1} u_i \left(\frac{t_{i+1} - t_{i-1}}{2} \right). \quad (1)$$

Therefore, the average speed of floating car r can be represented as $U_r = \frac{S_r}{t_p - t_0}$.

If the number of floating cars on the road section at a particular moment is n , the average speed of all cars on the road section can be written as $\tilde{U} = \frac{\sum_i^n U_r}{n}$.

- Speed distribution:

The car speed is distinguished into different levels and the histogram is used to represent the distribution of speed data of all floating cars on the read section. The standard of division is based on the speed distribution of floating cars in the city (Figure 3). Car speed data is mainly no more than 75 km/h except on a highway. Therefore, the car speed is distributed into 5 grades as shown in Table 2.

Table 2. The 5 levels of the speed of floating cars (km/h).

Speed Grades	1	2	3	4	5
Range(km/h)	<15	15~35	35~55	55~75	>75

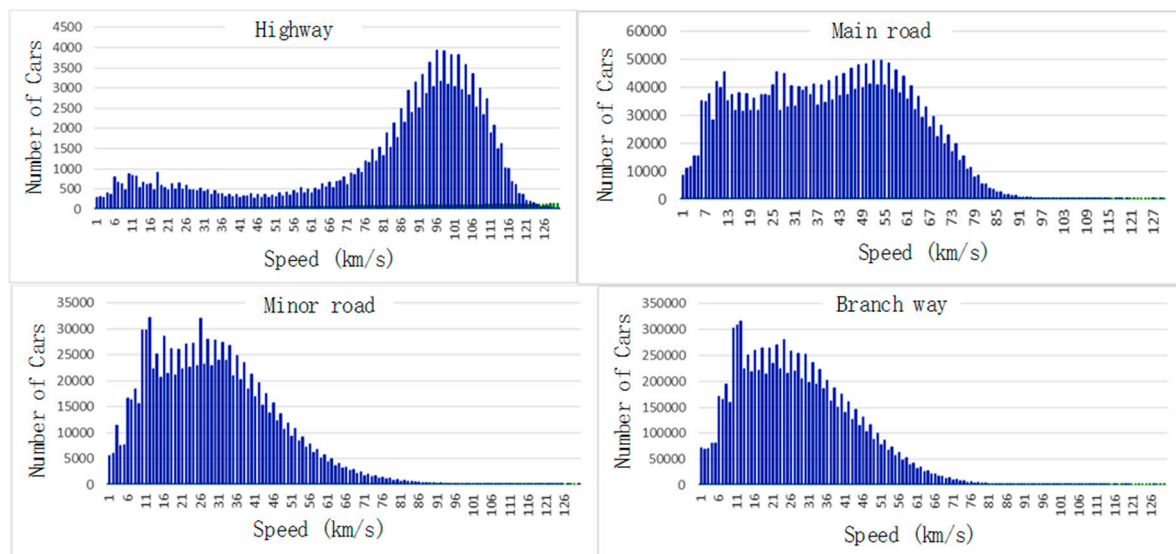


Figure 3. The speed distribution of floating cars in the city on 23 January 2015.

- Average stopping time:

When a car’s speed is below 5 km/h, it is identified as a stopped car. The stopping time of floating car r can be represented as $T_r = \sum_{i=0}^{p-1} (t_{i+1} - t_i)(u_i < 5)$.

Thus, the average stopping time of all cars on the road section can be written as $\tilde{T} = \frac{\sum_1^n T_r}{n}$.

From all kinds of eigenvalues above, 12-dimensional traffic congestion eigenvalue can be calculated.

2.3. Congestion Value

The work of labeling training samples is completed by five experts from the Transportation Department of the city. Through surveillance cameras, experts recorded information and gave evaluation of the traffic congestion at that time. Congestion evaluation is divided into three grades: Smooth, Average and Congested. The final label is in the grade which receives the most votes in 5 experts.

We indicate the congestion grades with the 3-dimensional eigenvalue. In each of the evaluation results, each dimension corresponds to a congestion level. The dimension of the corresponding congestion grade is set to 1 and other dimensions are set to 0. Thus, the congestion evaluation problem is transformed into a classification problem. Therefore, the evaluation result is the congestion grade which corresponds to the dimension having the largest value.

3. Kernel-Based SSELM

In a given training data set, the number of samples is n . A training sample (x, y) consists of congestion eigenvalue x and the corresponding congestion value y .

The formula of the single hidden layer feedforward neural network (SLFN) with N_h hidden nodes can be represented as:

$$\sum_{i=1}^{N_h} \beta_i g(w_i x + b_i) = y. \quad (2)$$

In a single hidden nodes i , g is the activation function, β_i is the output weight, w_i is the input weight, b_i is the bias. The output function of ELM can be written as:

$$y = F_{ELM}(x) = \sum_{i=1}^{N_h} \beta_i g(w_i x + b_i) = h(x)\beta, \quad (3)$$

where $h(x) = [g(w_1 x + b_1), \dots, g(w_{N_h} x + b_{N_h})]$ represents the output of hidden layer network, and $\beta = [\beta_1, \dots, \beta_{N_h}]^T$ represents the output weights of all hidden nodes. The output for all the n training samples can be written as:

$$Y = H\beta \quad Y = [y_1, \dots, y_n]^T \quad H = [h(x_1), \dots, h(x_n)]^T. \quad (4)$$

Previously, Huang et al. [8] used the smallest norm least squares solution of $Y = H\beta$. With a user-defined cost coefficient C , Huang et al. [10] optimized the calculation of the output weights β . Among them, when the number of hidden nodes is more than the number of training data, the result of β is:

$$\beta = H^T \left(\frac{I}{C} + HH^T \right)^{-1} Y. \quad (5)$$

Huang et al. [25] suggested to use a kernel function if the hidden layer feature mapping $h(x)$ is unknown. The kernel matrix χ for ELM can be written as follows, where $K(x_i, x_j)$ is Kernel function:

$$\chi_{ELM} = HH^T \quad \chi_{ELM_{i,j}} = h(x_i) \cdot h(x_j) = K(x_i, x_j). \quad (6)$$

Then the output function of Kernel-ELM can be represented as:

$$y = F_{ELM}(x) = h(x)\beta = h(x)H^T \left(\frac{I}{C} + HH^T \right)^{-1} Y = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_n) \end{bmatrix} \left(\frac{I}{C} + \chi_{ELM} \right)^{-1} Y. \quad (7)$$

Huang et al. [21] introduced manifold assumption into ELM, and proposed the solution of β in SSELM. In SSELM, Laplacian matrix is used to construct the distribution relationship among all training samples, so that both unlabeled and labeled data can be used to train the evaluation model in a united framework. For a training data set having l labeled samples and u unlabeled samples, the output weights β of a SSELM is:

$$\beta = H^T \left(I + \tilde{C}HH^T + \lambda LHH^T \right)^{-1} \tilde{C}\tilde{Y}. \quad (8)$$

The formula is valid when the number of hidden nodes is higher than the number of labeled samples l . The \tilde{Y} is the training target including the first l rows of labeled data equal to Y and the rest equal to 0. λ is user-defined semi-supervised learning rate. \tilde{C} is a $(l + u) \times (l + u)$ diagonal matrix with the first l diagonal elements of cost coefficient and the rest equal to 0. \tilde{C} can be calculated as:

$$C_i = \frac{C_0}{N_{P_i}} \quad i = 1, \dots, l, \quad (9)$$

where C_0 is user-defined cost coefficient, and N_{P_i} represents the sample quantity of the pattern of i th sample. L is Laplacian matrix, which can be calculated as $L = D - W$. $W = [w_{i,j}]$ is the similarity matrix of all the labeled and unlabeled samples. D is a diagonal matrix with its diagonal elements $D_{ii} = \sum_{j=1}^n w_{ij}$.

When the hidden layer is unknown, we use the Kernel function to calculate inner product without hidden layer feature mapping. The output function of Kernel-SSELM can be written as:

$$y = F_{SSELM}(x) = h(x)\beta = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_n) \end{bmatrix} \left(I + \tilde{C}\chi_{ELM} + \lambda L\chi_{ELM} \right)^{-1} \tilde{C}\tilde{Y}. \quad (10)$$

4. Evaluation Performance Results

4.1. Experimental Setup

In the experiment, we collect the floating car data from 15 June to 16 June 2015, and the quantity is more than 30,000,000 samples. The data is grouped in interval for 5 min and matched to the corresponding road section. Finally, we collect 13,681 samples. The evaluation of experts is based on the video from surveillance cameras at about 30 typical road sections in the city. In total, 537 valid samples were finally collected, and the remaining 13,144 samples were unlabeled. The experiment was implemented using Matlab R2013b (The MathWork Inc., Natick, MA, USA) on a 3.40 GHz machine with 4 GB of memory.

4.2. Comparisons with Related Algorithms

For comparison, we tested the empirical rule, SSELM, Kernel-SSELM and state-of-the-art semi-supervised learning algorithms such as TSVM [26], LDS [27], LapRLS [28], and LapSVM [28]. The test set had 100 samples randomly selected from the labeled sample, and the random generation process was repeated 10 times. The cost coefficient C_0 was fixed to 100 and the semi-supervised learning rate λ was chosen from in $\{2^{-20}, 2^{-19}, \dots, 2^{20}\}$. The kernel function of Kernel-SSELM is Gaussian function with the parameter γ fixed to 1. The number of hidden layer nodes of SSELM was set to 5000.

Table 3 shows that the evaluation model trained by Kernel-SSELM had the highest average accuracy at 86.2%. In addition, Kernel-SSELM only takes 48.2 s for training, which keep the high training efficiency of SSELM. In comparison, the other semi-supervised learning algorithms can also get relatively high accuracy, but the training consumptions are too huge.

Table 3. Evaluation result on realistic traffic data.

	Empirical Rule	TSVM	LDS	LapRLS	LapSVM	SSELM	Kernel-SSELM
Average Accuracy	68.9%	81.3%	82.2%	81.4%	84.8%	82.6%	86.2%
Best Accuracy	73.0%	87.0%	86.0%	86.0%	88.0%	87.0%	88.0%
Std. Dev.	3.79%	2.15%	2.71%	2.35%	1.97%	2.43%	1.55%
Training Time (s)	-	18,437	35,334	931	825	41.6	48.2

Figure 4 shows the confusion matrix of each evaluation model trained by machine learning methods in the 10 repetitions of the experiments. Geometric mean (G-mean) is used to evaluate the balance classification performance of each algorithm. In our work, the traffic evaluation is a multi-classification problem and the G-mean is calculated as:

$$G\text{-mean} = \left(\prod_{i=1}^k R_i \right)^{\frac{1}{k}} \quad R_i = \frac{TP_i}{N_i} \quad (11)$$

where R_i is the recall value of each class, N_i is the number of samples in the class, and TP_i is the number of correctly classified samples of the class. k is the number of class, and it was set as 3 in our work.

As seen from the Figure 4, each method has a certain degree of unbalanced classification problem. The G-mean is about 2% or more lower than the accuracy. The majority class “Smooth” is easier to distinguish, and the minority classes “Average” and “Congested” have lower recall value. There are many samples incorrectly classified between classes “Average” and “Congested”, which means these two classes are more difficult to distinguish. Compared to the other method, Kernel-SSELM can more accurately classify the class “Smooth”, so that it gets higher accuracy and G-mean.

Evaluated Class	Actual Class		
	Smooth	Average	Congested
Smooth	479	55	0
Average	60	224	36
Congested	0	36	110
Accuracy	81.3%	G-mean	78.1%

(a)

Evaluated Class	Actual Class		
	Smooth	Average	Congested
Smooth	476	53	0
Average	63	236	36
Congested	0	26	110
Accuracy	82.2%	G-mean	79.3%

(b)

Evaluated Class	Actual Class		
	Smooth	Average	Congested
Smooth	478	51	0
Average	61	224	34
Congested	0	40	112
Accuracy	81.4%	G-mean	78.5%

(c)

Evaluated Class	Actual Class		
	Smooth	Average	Congested
Smooth	485	43	0
Average	54	249	32
Congested	0	23	114
Accuracy	84.8%	G-mean	82.2%

(d)

Evaluated Class	Actual Class		
	Smooth	Average	Congested
Smooth	477	46	0
Average	62	237	34
Congested	0	32	112
Accuracy	82.6%	G-mean	79.9%

(e)

Evaluated Class	Actual Class		
	Smooth	Average	Congested
Smooth	493	37	0
Average	46	255	32
Congested	0	23	114
Accuracy	86.2%	G-mean	83.3%

(f)

Figure 4. Confusion matrix and G-mean of each machine learning algorithms: (a) TSVM; (b) LDS; (c) LapRLS; (d) LapSVM; (e) SSELM; (f) Kernel-SSELM.

Figure 5 shows that with the increase of the number of unlabeled data in the training set, the performance of Kernel-SSELM and SSELM get better. In particular, Kernel-SSELM gives a more significant growth when the number of unlabeled parts is below 40%, and its performance becomes stable when there are more unlabeled data. By comparison, the performance of SSELM grows slower and it is hard to get stable result.

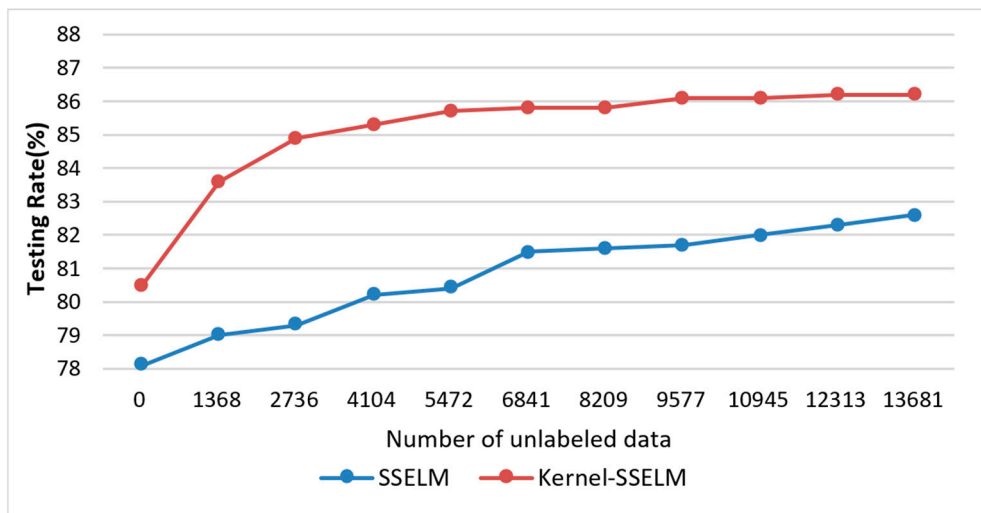


Figure 5. Evaluation comparison with different number of unlabeled data.

4.3. Performance Sensitivity on Parameters

Figure 6 shows the performance sensitivity. In order to achieve good generalization performance, the cost coefficient C_0 and kernel parameter γ need to be chosen appropriately. The best performance is usually achieved in a very narrow range of cost coefficient C_0 , which means the Kernel-SSEL model is very sensitive to C_0 . However, when the C_0 is properly chosen, the performance will be not sensitive to the kernel parameter γ . Semi-supervised learning rate λ also significantly affects the performance of Kernel-SSEL. There is an obvious trend that Kernel-SSEL gives a lower prediction error with more unlabeled data. However, the best performance is achieved in a very narrow range of λ . Moreover, with different numbers of unlabeled data, the best value of λ varies in a small range.

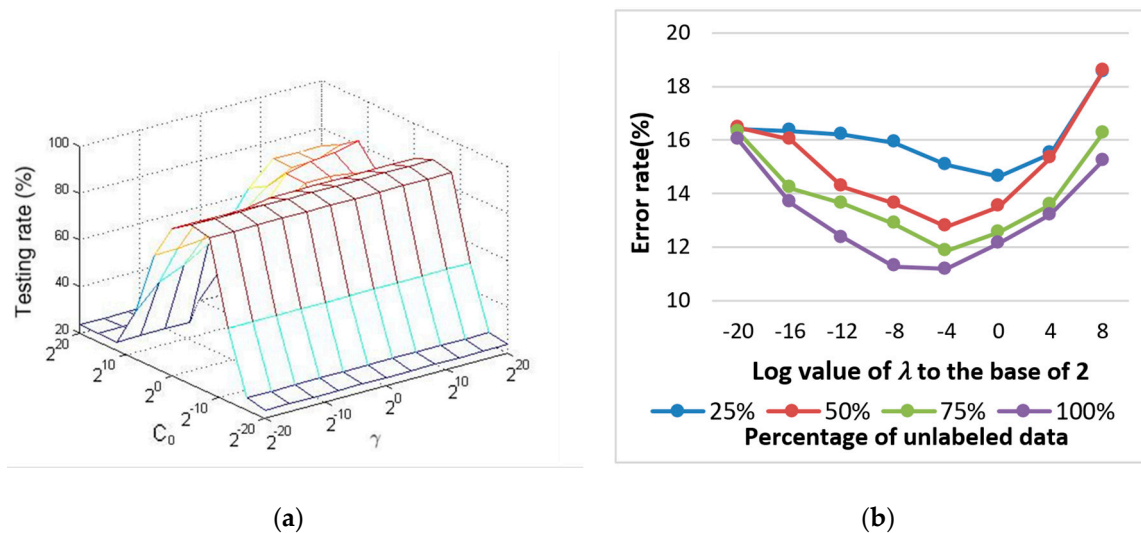


Figure 6. Performance sensitivity of Kernel-SSEL on the parameters: (a) The sensitivity of cost coefficient C_0 and kernel parameter γ ; (b) The sensitivity of semi-supervised learning rate λ .

5. Evaluation on the Realistic Traffic Data

The trained model was used in the Urban Transportation Assessment and Forecast System in the city of Nanning, China. The evaluation results are shown on the map through the website.

Figure 7 displays the real-time traffic condition evaluation. In the map, green represents smooth traffic, yellow shows average condition, and red means the road is congested.



Figure 7. Real-time traffic evaluation.

Besides the quantitative analysis of evaluation accuracy, it is also important to investigate whether the evaluation results are consistent with the intuitive feeling of citizens. Figure 8 shows the comparison between the evaluation on the map and the real road conditions taken by surveillance cameras. Seen from the Figure 8, the traffic evaluation accurately reflects the road traffic congestion at that time.

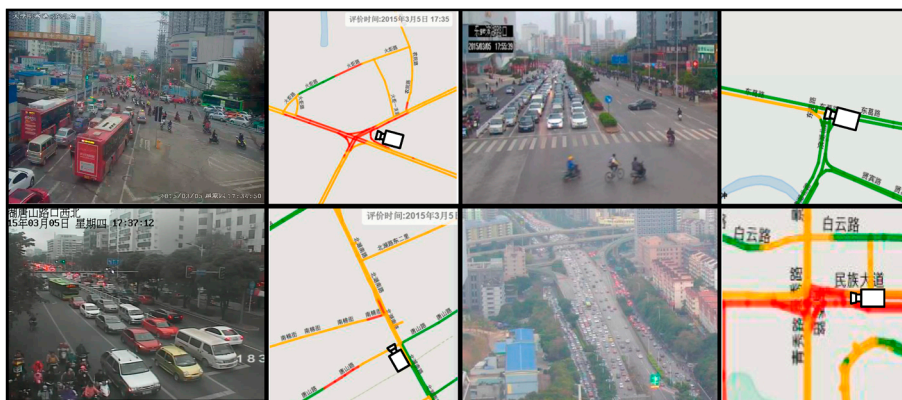


Figure 8. Performance of real-time traffic evaluation on road level.

The Urban Transportation Assessment and Forecast System also use thermodynamic chart to provide evaluation on zone level. In the Figure 9, the deeper the color, the more serious the congestion of a zone.

Seen from the map, the congestion concentrates in the center of the city including the central business circle, train station and central main road. These places are in the old city district, which has small network capacity, large population and big traffic flow, so the congestion is more serious. In addition, beltway entrances, main road overpass bridges and rail transit nodes also have concentrated traffic. Comparing the traffic condition between morning and evening, congestion is more serious at 18:00, which means traffic is more concentrated in the evening peak. These states are consistent with the actual situation of urban traffic in Nanning. Therefore, based on the congestion accurate evaluation, the transportation assessment system has high reliability and practicability.

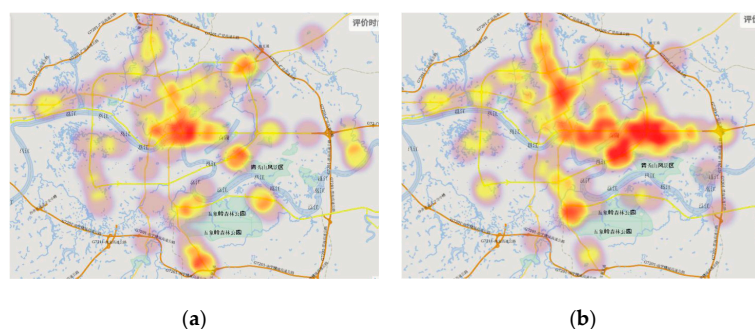


Figure 9. Real-time traffic evaluation on zone level: (a) Congestion evaluation at 8:00 in the morning; (b) Congestion evaluation at 18:00 in the evening.

6. Conclusions

In this paper, semi-supervised learning solved the asymmetry problem between traffic data quantity and unit information content. Our method was used to train the evaluation model by integrating small-scale labeled data and large-scale unlabeled data cooperatively. The extreme learning machine has high training efficiency and is easy to implement. In the case of large data scales, high training speed ensures that, despite changing traffic conditions, it can still renew training several times to choose a better model. Kernel function improves the recognition accuracy. In neglecting the number of hidden layer nodes, the optimization of kernel function improves the stability. Both the experiment and the real-time application show that the evaluation system builds a connection between congestion condition and road information with high efficiency and accuracy. The evaluation system clearly shows the congestion on the GIS map, which can provide assistance for traffic control and public travel.

Acknowledgments: This work was supported by National Nature Science Foundation of P. R. China (No. 61572075) and National Key Research and Development Program of China (No. 2016YFB0700502, 016YFB1001404), which covering the costs to publish in open access.

Author Contributions: Qing Shen and Xiaojuan Ban conceived the proposed method and designed the traffic congestion system; Qing Shen and Chong Guo performed the experiments and the system implementation; Qing Shen and Chong Guo analyzed the data; Qing Shen wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Gu, J.; Zhou, S.H.; Yan, X.P.; Deng, L.F. Formation Mechanism of Traffic Congestion in View of Spatio-temporal Agglomeration of Residents' Daily Activities: A Case Study of Guangzhou. *Sci. Geogr. Sin.* **2012**, *32*, 921–927.
2. Wen, H.; Sun, J.; Zhang, X. Study on Traffic Congestion Patterns of Large City in China Taking Beijing as an Example. *Procedia Soc. Behav. Sci.* **2014**, *138*, 482–491. [[CrossRef](#)]
3. Liu, R.; Hu, W.P.; Wang, H.L.; Wu, C. The Road Network Evolution Analysis of Guangzhou-Foshan Metropolitan Area Based on Kernel Density Estimation. In Proceedings of the International Conference on Computational and Information Sciences, Chengdu, China, 17–19 December 2010; pp. 316–319.
4. Kong, Q.J.; Zhao, Q.; Wei, C.; Liu, Y. Efficient traffic state estimation for large-scale urban road networks. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 398–407. [[CrossRef](#)]
5. Batool, F.; Khan, S.A. Traffic estimation and real time prediction using Ad Hoc networks. In Proceedings of the IEEE Symposium on Emerging Technologies, Catania, Italy, 18 September 2005; pp. 264–269.
6. Gomez, A.E.; Alencar, F.A.R.; Prado, P.V.; Osorio, F.S.; Wolf, D.F. Traffic lights detection and state estimation using hidden markov models. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 8–11 June 2014; pp. 750–755.

7. Purusothaman, S.B.; Parasuraman, K. Vehicular traffic density state estimation using Support Vector Machine. In Proceedings of the 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), Tirunelveli, India, 25–26 March 2013; pp. 782–785.
8. Huang, G.B.; Siew, C.K. Extreme learning machine: RBF network case. In Proceedings of the 2004 Control, Automation, Robotics and Vision Conference (ICARCV), Kunming, China, 6–9 December 2004; Volume 2, pp. 1029–1036.
9. Huang, G.B.; Chen, L. Letters: Convex incremental extreme learning machine. *Neurocomputing* **2007**, *70*, 3056–3062. [[CrossRef](#)]
10. Rong, H.J.; Huang, G.B.; Sundararajan, N.; Saratchandran, P. Online Sequential Fuzzy Extreme Learning Machine for Function Approximation and Classification Problems. *IEEE Trans. Syst. Man Cybern. Part B* **2009**, *39*, 1067–1072. [[CrossRef](#)] [[PubMed](#)]
11. Kasun, L.L.C.; Zhou, H.; Huang, G.B.; Vong, C.M. Representational learning with extreme learning machine for big data. *IEEE Intell. Syst.* **2013**, *28*, 31–34.
12. Yang, Y.; Wu, Q.M.J. Multilayer Extreme Learning Machine with Subnetwork Nodes for Representation Learning. *IEEE Trans. Cybern.* **2015**, *46*, 2570–2583. [[CrossRef](#)] [[PubMed](#)]
13. Baradarani, A.; Wu, Q.M.J.; Ahmadi, M. An efficient illumination invariant face recognition framework via illumination enhancement and DD-DTCWT filtering. *Pattern Recognit.* **2013**, *46*, 57–72. [[CrossRef](#)]
14. Yang, Y.; Wu, Q.M.J.; Wang, Y. Autoencoder With Invertible Functions for Dimension Reduction and Image Reconstruction. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *99*, 1–15. [[CrossRef](#)]
15. Heeswijk, M.V.; Miche, Y.; Oja, E.; Lendasse, A. GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing* **2011**, *74*, 2430–2437. [[CrossRef](#)]
16. Cao, J.; Lin, Z.; Huang, G.B.; Liu, N. Voting based extreme learning machine. *Inf. Sci.* **2012**, *185*, 66–77. [[CrossRef](#)]
17. Luo, J.; Vong, C.M.; Wong, P.K. Sparse Bayesian Extreme Learning Machine for Multi-classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 836–843. [[PubMed](#)]
18. Bai, Z.; Huang, G.B.; Wang, D.; Westover, M.B. Sparse extreme learning machine for classification. *IEEE Trans. Cybern.* **2014**, *44*, 1858–1870. [[CrossRef](#)] [[PubMed](#)]
19. Cao, J.; Lin, Z. Extreme learning machines on high dimensional and large data applications: A survey. *Math. Probl. Eng.* **2015**, *2015*, 1–14. [[CrossRef](#)]
20. Cao, J.; Xiong, L. Protein sequence classification with improved extreme learning machine algorithms. *BioMed Res. Int.* **2014**, *2014*, 1–14. [[CrossRef](#)] [[PubMed](#)]
21. Huang, G.; Song, S.; Gupta, J.N.D.; Wu, C. Semi-supervised and unsupervised extreme learning machines. *IEEE Trans. Cybern.* **2014**, *44*, 2405–2417. [[CrossRef](#)] [[PubMed](#)]
22. Pal, M.; Maxwell, A.E.; Warner, T.A. Kernel-based extreme learning machine for remote-sensing image classification. *Remote Sens. Lett.* **2013**, *4*, 853–862. [[CrossRef](#)]
23. Liu, H.; Qin, J.; Sun, F.; Guo, D. Extreme Kernel Sparse Learning for Tactile Object Recognition. *IEEE Trans. Cybern.* **2016**. [[CrossRef](#)] [[PubMed](#)]
24. Shen, Q.; Ban, X.; Guo, C.; Wang, C. Kernel Semi-supervised Extreme Learning Machine Applied in Urban Traffic Congestion Evaluation. In Proceedings of the 13th International Conference on Cooperative Design, Visualization, and Engineering (CDVE), Sydney, Australia, 24–27 October 2016; pp. 90–97.
25. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B* **2012**, *42*, 513–529. [[CrossRef](#)] [[PubMed](#)]
26. Joachims, T. Transductive inference for text classification using support vector machines. In Proceedings of the 6th International Conference on Machine Learning (ICML), Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 200–209.
27. Drigo, M.; Maniezzo, V.; Colorni, A. The ant system: Optimization by a colony of cooperation agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 1–13. [[CrossRef](#)] [[PubMed](#)]
28. Belkin, M.; Niyogi, P.; Sindhvani, V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *J. Mach. Learn. Res.* **2006**, *7*, 2399–2434.

