Locality Constraint Distance Metric Learning for Traffic Congestion Detection

Qi Wang, Jia Wan, Yuan Yuan

School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.

Abstract

In this paper, a locality constraint distance metric learning is proposed for traffic congestion detection. First of all, an accurate and unified definition of congestion is proposed and the congestion level analysis is treated as a regression problem in the paper. Based on that definition, a dataset consists of 20 different scenes is constructed for the first time since there exists no traffic congestion dataset containing multiple scenes. To characterize the congestion level in different scenes, the low-level texture feature and Kernel Regression is utilized to detect traffic congestion level. To reduce the influence among different scenes, a Locality Constraint Distance Metric Learning (LCML) which ensured the local smooth and preserved the correlations between samples is proposed. The extensive experiments confirm the effectiveness of the proposed method.

Keywords: Distance metric learning, locality constraint, kernel regression, traffic congestion analysis

1. Introduction

As the development of the society, the traffic has become more and more congested around the world. The traffic jams not only waste our time and resources, but also create more pollutions and accidents. If the traffic congestion level can be automatically detected, it will be easier to relieve the traffic jams.

There are three popular devices that can be utilized to detect congestion level. The first one is Loop Detector [1] whose installation and maintenance of the devices is complicated and the detection range is very limited. The second one is GPS based smart cell phone and vehicle. This method is popular because of the wide detection range, but the precision of the congestion level detection is low. The last one is camera. The cameras are very popular nowadays, so, the detection range could be ensured. What's more, the detection could be very precise.

Many approaches are proposed to detect congestion level from videos [2, 3]. Most of them detect congestion by analyzing the key points or moving areas. The number of key points or moving blobs and their speed can be used to estimate the congestion. These methods are effective to answer whether it is a traffic jam but can't determine the exact congestion level. Besides, these methods solve this problem in only one specific scene which is not useful for real applications.

Unfortunately, the research of traffic congestion analysis is limited because there exist many problems. The first one is that the researchers treated congestion detection as a classification task which simplifies the problem but limits its usage in real applications. The second is that there exists no large scale dataset containing different scenes for traffic congestion analysis. The most challenge is the different illumination, occlusion level, camera angles and road conditions in various scenes. These problems make the traffic congestion status hard to analyze.

To remedy these problems, an accurate and unified definition is first proposed. With this definition, the congestion analysis becomes a regression problem since the label of congestion level is a real number between 0 to 1. At the same time, a dataset consists of 20 different scenes is conducted to serve as the platform for congestion analysis.

To reduce the effect of different conditions which makes the congestion level hard to analyze, the metric learning [4, 5] is utilized. However, traditional metric learning for regression predicts the value of a sample by measuring its distance to all the other training samples. But, the performance is bad since the generalization ability is limited and different scenes will affect the prediction.

Thus, a locality constraint metric learning is proposed in which only several nearest neighbors are considered.

The contributions of the paper are summarized as follows:

- An accurate and unified definition of congestion is proposed. With the definition, the question of how congested the traffic status is could be answered. Besides, the congestion level in different scenes can be compared in a unified framework.
- A dataset consisting of 20 different scenes is constructed for the first time to promote the research of traffic congestion analysis. The videos in this dataset contain different illumination, camera angles and road conditions. This dataset can serve as a platform for the research of congestion level regression.
- A locality constraint metric learning is proposed for congestion regression. Since the difference among various scenes affects the performance of congestion regression, this approach is proposed to reduce the defect across different scenes through constraining that only the neighbors of a testing sample can contribute to the prediction.

The remaining part of this paper is as follows. Relevant works are reviewed in Section 2 and the definition and dataset are presented in Section 3. Then, the details of the proposed method are elaborated in Section 4. After the experimental results are reported and discussed in Section 5, the conclusion and future works are presented in Section 6.

2. Related Work

In this section, the relevant works of congestion detection and metric learning are briefly reviewed.

2.1. Congestion Detection

The congestion detection algorithms can be divided into two categories. The first category is based on the analysis of key points and moving areas. Another is based on direct feature extraction and classification.

The assumption of first category is that more congested traffic scenes contain more moving objects. Hu *et al.* [6] proposed an algorithm that classifies congestion videos based on the segmentation of moving vehicles. Firstly, the moving objects are segmented by background subtraction method [7]. Then, the speed of the moving blobs are calculated by Optical Flow [8]. Finally, the percentage of moving blobs and their speed can be served as the features. Fuzzy logic is utilized for the final decision. Sobral *et al.* [9] proposed an approach which combined the features of key points and moving blobs together. The crowd density is first evaluated by background subtraction algorithm. Then, the speed is estimated by Kanade-Lucas-Tomasi (KLT) algorithm [10]. These methods is rely on the preprocessing like Background Subtraction and Tracking. Thus, the performance of these congestion detection methods is limited because of the uncertain preprocessing.

The purpose of the other category is to design congestion related features. Derpanis *et al.*[11] proposed the Spatialtemporal Orientation Analysis feature motivated by the visual dynamics of congested scenes. Riaz *et al.*[12] encoded motion information by analyzing the statistics of motion vectors. To combine the appearance and motion information together, Dallalzadeh *et al.* [13] proposed the symbolic representation. These methods don't rely on the preprocessing algorithms which make them work well in a specific scene. However, how to design the features that can cross congested scenes is still a challenging task.

2.2. Metric Learning

Metric Learning is the task to learn a good distance measurement. The aim of the metric learning is to minimize the distance of samples from the same class and maximize the distance from different classes [14].

Most of the distance learning algorithms are designed for classification task [15, 16], such as image retrieval [17], person Re-identification [18] and face recognition [19, 20]. Large Margin Nearest Neighbor (LMNN) [21, 22] is proposed to learn a Mahalanobis distance for K Nearest Neighbor (KNN) classification. Information-Theoretic Metric Learning (ITML) [23] is also proposed for KNN classification. The difference is that the distribution parameterized by distance metric M is regularized to be closed to a prior distribution. Neighborhood Component Analysis (NCA) [24] is another linear metric learning algorithm which optimized the classification performance based on the Leave-one-out validation. To exploit the negative constraint lacking in RCA, Discriminative Component Analysis (DCA) [25] is proposed.

The metric learning can also be included into regression task. Metric Learning for Kernel Regression (MLKR) [26] is proposed to learn a distance metric for kernel regression. To combine the sparsity into the framework, Kernel Regression with Sparse Metric Learning (KRSML) [27] is proposed to regularized the distance metric with a mixed (2,1)-norm. Xiao *et al.* [28] proposed an application which utilized metric learning for human age estimation.

Recently, many deep metric learning methods are proposed with the development of deep learning. Hu *et al.* [29] proposed an deep metric learning method to compare the similarity of two faces by minimizing the intra-class variation and maximize the inter-class variation. Li *et al.* [30] proposed to learn a suitable metric with the help of community-contributed images. Song *et al.* [31] proposed an novel structural objective function on the lifted problem which is proved to be effective for image retrieval.

3. The Definition and Dataset

In this section, a unified and accurate definition is first presented. And then, the detail of the traffic congestion dataset and the corresponding labeling method are introduced.

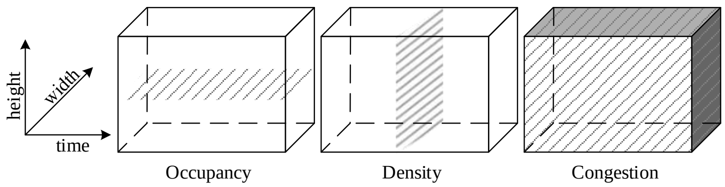

Figure 1: The proposed definition is based on the time-space congestion.

3.1. Definition

The congestion level is define as the occupancy of the moving objects in the domain of space-time. In literature, the congestion can be measured by spatial congestion and temporal congestion which called density and occupancy respectively [1]. As shown in Figure 1, the density can only measures the congestion at a point of time, while the occupancy can only measures the congestion at a point of space. The proposed definition of congestion considers the spatial and temporal information simultaneously This definition is accurate and unified, so the comparison of congestion level in different scenes becomes possible.

Formally, the congestion can be expressed as follows:

$$congestion = \frac{\sum\limits_{x,y,t} f(x,y,t)}{width \times length \times time} \tag{1}$$

where $congestion \in (0,1)$ is the congestion level, and $time$ is the number of frames in a video clip. $width$ and $length$ are the width and length of a road. $f(x,y,t)$ is defined as:

$$f(x,y,t) = \begin{cases} 1, & \text{occupied} \\ 0, & \text{not occupied} \end{cases} \tag{2}$$

which indicates that whether a point is occupied by a moving object.

3.2. Traffic Congestion Dataset

Since there exists no dataset for cross scene congestion level detection, a new dataset which consists of 20 different scenes is constructed. First, large amount of videos containing different streets and weather conditions are collected. The

Figure 2: Typical images of the traffic congestion dataset.

resolution of these videos are equal or larger than 1080×720. The average length of these videos is 30 minutes. Different direction on same road is treated as different scenes. Typical images are shown in Figure 2

With the definition, the calculation of the real congestion level needs a pixel-wise labeling which is time-consuming. To remedy this, we suppose that vehicles and lanes have the same width. With this assumption, the congestion level can be reduced as follows:

$$congestion = \frac{\sum\limits_{y,t} f(y,t)}{height \times time} \tag{3}$$

where *height* can be seen as the length of the road.

With the simplification, the moving objects and road can be represented as a line along them and the congestion level can be calculated easily. The visualization of labeling is shown in Figure 3. Specifically, the length of lines on the road is represented by the length of a line along it. Similarly, the length of vehicles can be estimated in the same way. Then, the congestion level can be calculated by the fraction of total length of vehicles and the total length of lanes.

Since the perspective transformation can be heavily affect the real congestion level, it is considered in the labeling procedure. The perspective transformation causes that the vehicles far from the camera will be smaller than the vehicles close to the camera. Motivated by this, the weight of pixels at the top of images (where the vehicles are far from the camera) should be larger. After the perspective is considered, the variation of the congestion level caused by perspective transformation is reduced. As shown in Figure 4, the congestion level is more smooth after the perspective transformation is taken into consideration.

4. Our Method

In this section, the details of the proposed method are presented. First of all, the texture feature of image is extracted as low-level features. Then, a distance metric is learned by the proposed Locality Constraint Metric Learning algorithm with the precomputed features. Based on the learned metric, the congestion level can be efficiently detected by Kernel Regression.

4.1. Locality Constraint Distance Metric Learning

The influence among different scenes makes the prediction of congestion level a hard task. To reduce the effect of different scenes, a locality constraint distance metric learning is proposed. This method only takes the neighbors of the testing sample into consideration, since the testing sample and its neighbors has high probability of belonging to the same scene. That reduces the effect among scenes efficiently.

The Kernel Regression can be treated as the weighted sum of training samples. The weight should be related to the similarity between samples. Formally, given a feature vector x_i of a sample, the corresponding congestion level $\hat{y}_i$ can

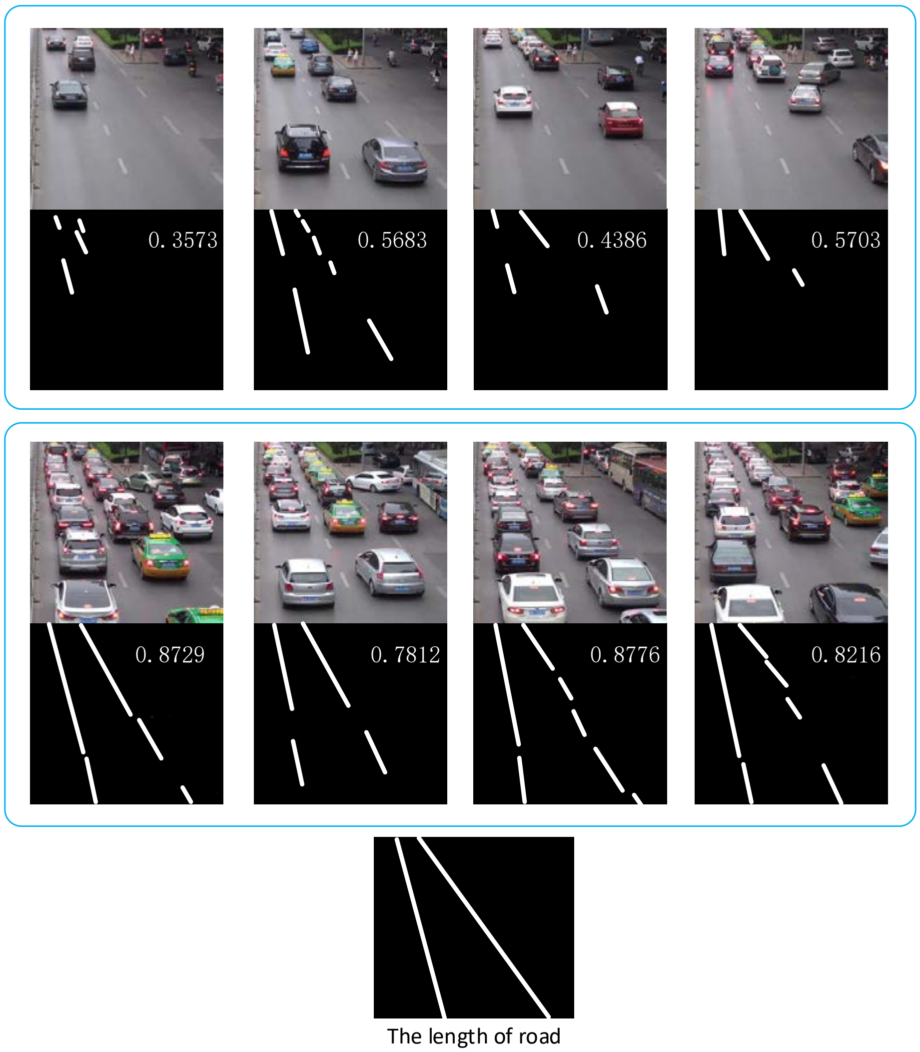

Figure 3: The visualization of labeling method. In this figure, two different scenes is used for examples. The numbers on the binary image indicate the congestion level. In the proposed labeling method, each vehicle can be represented by a line along it. Since the there are two lanes in the road, the length of road is represented by two lines as the bottom image shows.

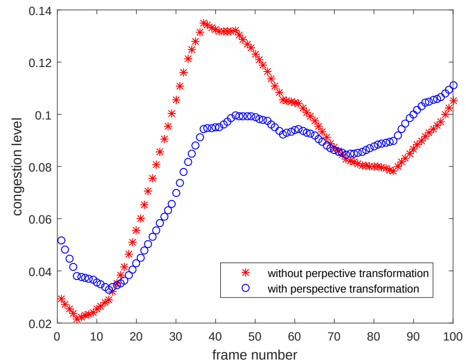

Figure 4: The congestion level is more smooth after the perspective transformation is taken into consideration. In this figure, the horizontal axis is the frame number and the vertical axis is the congestion level. The red stars are congestion level without perspective transformation. The blue circles are congestion level with perspective transformation.

be calculated as:

$$\hat{y_i} = \frac{\sum_{j \neq i} y_i k_{ij}}{\sum_{j \neq i} k_{ij}}, \tag{4}$$

where k_{ij} refers to the kernel function which is defined as:

$$k_{ij} = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{d_{ij}}{\sigma}), \tag{5}$$

where d_{ij} is the distance of x_i and x_j. Note that, σ is fixed to 1 for simplification. Usually, the Euclidean distance is included as the distance measurement. However, many works have shown that the learned metric can generate better performance. Thus, the distance metric learning is included to learn a better distance measurement.

In metric learning, the distance between two vector x_i and x_j is calculated as follows:

$$d_{ij} = (x_i - x_j)^\top M (x_i - x_j), \tag{6}$$

where M is the learned distance metric that can transform the features to learned space and produce better performance. Note that M needs to be preserved to be semi-definite, which is hard to satisfy. Motivated by [26], M can be decomposed to $A^\top A$. Then, equation 6 can be reformulated as follows:

$$d_{ij} = \|A(x_i - x_j)\|^2. \tag{7}$$

To learn the distance metric L, the mean squared error between the ground truth and the prediction can be used as the loss function:

$$\mathcal{L}_{mse} = \sum_{i} \left((y_i - \hat{y_i})^2 + \beta \sum_{j \neq i} (k_{ij} \times d_{ij}) \right), \tag{8}$$

where y_i is the ground truth and $\hat{y_i}$ is the prediction. d_{ij} is the distance of two samples and k_{ij} can be treated as the weight in the Kernel Regression in Equation 4. This regularization term is used to punish the weights of x_j that is far from x_i.

To ensure the divergence invariant under scaling of the feature space, the LogDet divergence [23] is utilized. Then, the final loss function is as belows:

$$\mathcal{L} = \mathcal{L}_{mse} + \beta D_{ld}(A^\top A, M_0), \tag{9}$$

where M_0 is the prior metric which is set as identity matrix in our experiments. The $D_{ld}(M, M_0)$ is defined as:

$$D_{ld}(M, M_0) = \mathbf{tr}(MM^{-1}) - \log\det(MM_0^{-1}) - d. \tag{10}$$

Motivated by [32], $D_{ld}(M, M_0)$ can be replaced by:

$$D_{ld}(M, M_0) = \mathbf{tr}(MM^{-1}) - \log\det(M). \tag{11}$$

4.2. Approximation and Optimization

To minimize the optimization problem in Equation 8, an approximation of the problem is proposed. The aim of Equation 8 is to find out some neighbors of sample x_i in the training set and give a correct prediction through the weighted average of these neighbors. That suggests we can use K neighbors for prediction instead of all samples. This greatly reduced the influence among different scenes. Thus, Equation 1 can be reformulated as:

$$\hat{y_i} = \frac{\sum_{j \in N(i)} y_i k_{ij}}{\sum_{j \in N(i)} k_{ij}}. \tag{12}$$

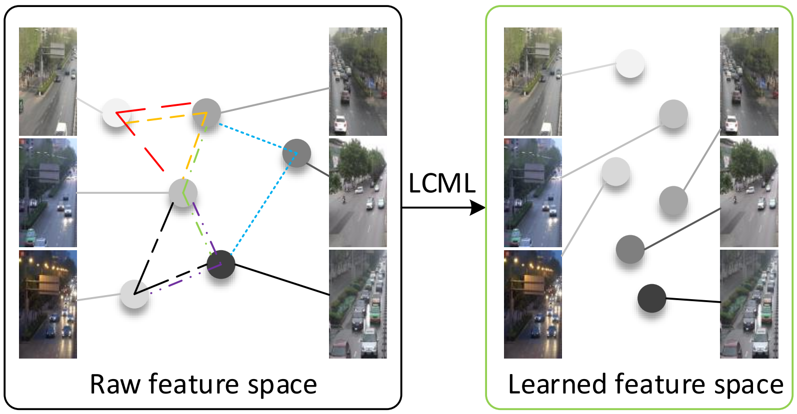

Figure 5: The proposed metric learning algorithm constrains that the value of a sample can only be approximated by its neighbors, which minimizes the influence among different scenes. In this Figure, Each point represent the feature of the congestion level in the feature space. In LCML, only neighbors are considered for prediction as the dotted lines show.

where $N(i)$ is the set of neighbors of x_i. Consequently, $\mathcal{L}_{mse}$ loss can be reduced as follows:

$$\mathcal{L}_{mse} = \sum_i (y_i - \hat{y}_i)^2. \tag{13}$$

This final problem can be efficiently solved by gradient decent algorithms and preserve the locality at the same time. To solve this optimization problem, the gradient of $\mathcal{L}$ with respect to A is calculated as follows:

$$\frac{\partial \mathcal{L}}{\partial A} = \frac{\partial \mathcal{L}_{mse}}{\partial A} + \beta \frac{\partial \mathcal{D}_{ld}(A^\top A, M_0)}{\partial A}, \tag{14}$$

where $x_{ij} = x_i - x_j$ and

$$\frac{\partial \mathcal{L}_{mse}}{\partial A} = 4A \sum_i (\hat{y}_i - y_i) \sum_{j \in N(i)} (\hat{y}_j - y_j) k_{ij} x_{ij} x_{ij}^\top, \tag{15}$$

and

$$\frac{\partial D_{ld}(AA^\top, M_0)}{\partial A} = 2A(M_0^{-1} + (A^\top A)^{-1}) \tag{16}$$

The details of the gradient decent procedure are shown in Algorithm 1. In this algorithm, the X is the feature matrix containing n samples in $\Re^d$ and Y is the corresponding congestion level. The learned transformation A is obtained by minimizing the loss function $\mathcal{L}$ defined in Equation 9.

Algorithm 1 Locality constraint metric learning

Input: X: feature matrix containing n samples in $\Re^d$;

Y: n corresponding congestion level.

repeat

Calculate $\nabla A \leftarrow \frac{\partial \mathcal{L}}{\partial A}$ via Equation 14

Initialize $A_{best} \leftarrow 0$, $\mathcal{L}_{best} \leftarrow 0$

Calculate $A' \leftarrow A - \delta \nabla A$

Calculate $\mathcal{L}' \leftarrow \mathcal{L}(A')$ via Equation 9

if $\mathcal{L}' < \mathcal{L}_{best}$ **then**

Update A_{best}: $A_{best} \leftarrow A'$

end if

until A is convergence.

Output: The learned transformation A.

5. Experiments

In this section, extensive experiments are conducted to confirm the effectiveness of the proposed method. Firstly, the experimental parameters are selected through cross-validation. Then, the proposed method is compared to some traditional algorithms. After that, the effectiveness of the feature and classifier is evaluated.

5.1. Parameter Settings

The constructed dataset consists of 20 different scenes which has different lights, occlusions and road conditions. The average length of these videos is 30 minutes. The resolutions are 1280×720 and 1920×1080. Typical images are shown in Figure 2.

In the experiments, 5000 samples are used for training, 1500 for validation and 1500 for testing. The number of neighbors K is set as 100 and 10 for training and testing as shown in Figure 6. Note that the Mean Squared Error (MSE) is employed for the evaluation. The lower MSE indicates better performance. The

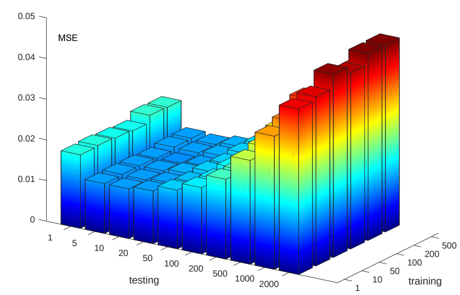

Figure 6: The axises are K in the training, K in the testing, and Mean Squared Error. For the best performance, K is set as 100 and 10 for training and testing. Note that the lower MSE indicates better performance.

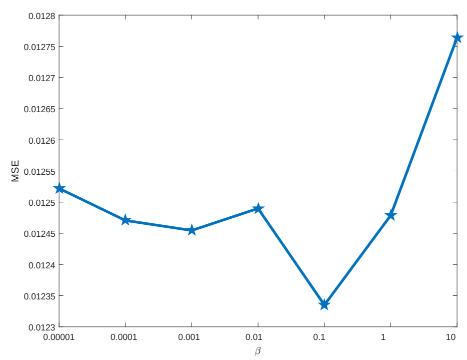

Figure 7: The selection of parameter β. In this figure, the horizontal axis is the different β and the vertical axis is the Mean Squared Error. Note that the lower MSE indicates better performance.

parameter β in Equation 9 is set as 0.1 through cross validation. The result is shown in Figure 7.

5.2. The evaluation of the proposed method

In this section, the proposed method is compared to some traditional algorithms. The comparison methods are Linear Regression (LR), Kernel Regression (KR) [33] and Metric Learning for Kernel Regression (MLKR) [26]. In this experiment, the low-level texture feature is utilized as the representation of congestion level. The final results comparison is shown in Table 1.

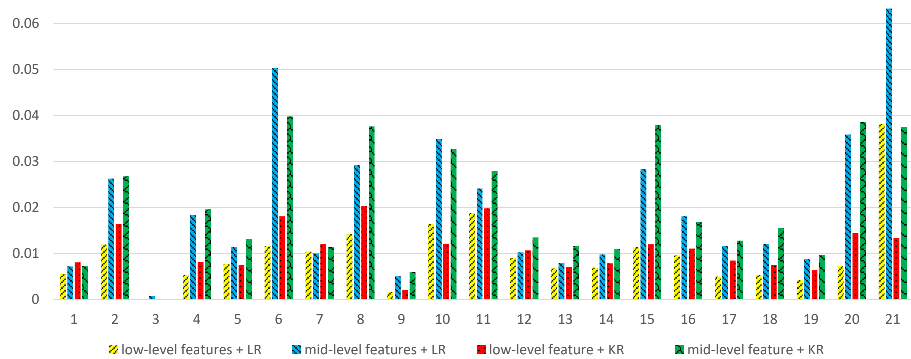

Figure 8: The comparison of different features and classifiers. In this figure, $1-20$ of the horizontal axis are the results of the experiments in which the training and testing are performed on only one scene. 21 of the horizontal axis is the result of the experiment in which the whole dataset (20 different scenes together) is included for training and testing.

As shown in Table 1, the performance of Kernel Regression is better than Linear Regression, which indicates that a non-linear classifier outperforms a linear classifier for congestion regression. It is unsurprising since the cross scenes congestion level detection is a non-linear regression problem. However, it is surprising that Metric Learning for Kernel Regression is not superior to Kernel Regression. The result indicates that Metric Learning is ineffective under the influence of different scenes. After the locality constraint is added to the loss function, Metric Learning shows its superiority. Thus, the influence among different scenes will affect the prediction of Metric Learning for Kernel Regression, and locality constraint can reduce that influence efficiently.

The reason why Locality Constraint Metric Learning outperforms other methods is that the locality is preserved during the learning of distance metric. The locality is guaranteed by constraining that only neighbors of a sample can be included for prediction. With this restraint, the similar congestion scenes will have similar prediction since their neighbors should be similar as well.

Methods	LR	KR	MLKR	LCML
MAE	0.104	0.083	0.089	**0.076**
MSE	0.038	0.013	0.014	**0.010**

Table 1: Comparison of Linear Regression (LR), Kernel Regression (KR), Metric Learning for Kernel Regression (MLKR) and the proposed Locality Constraint Distance Metric Learning (LCML). MAE is mean absolute error and MSE is mean squared error.

5.3. The Effectiveness of Feature and Classifier

To confirm that the feature is useful for congestion regression, the mid-level features [34] are included as the comparison feature. The extraction of mid-level feature is consists of dictionary learning and feature encoding. Specifically, the Batch K-means [35, 36] is included for dictionary learning and the sparse coding [37] is utilized to encode an image into final descriptor. The results are shown in Figure 8.

First of all, if the training and testing is performed on only one scene, the performance is better than the training and testing across different scenes. As shown in Figure 8 that the most errors of $1-20$ (single scene, and remind that the constructed dataset contains 20 different scenes) is lower than the error of 21 (multiple scenes). That confirms the influence of different scenes will drop the congestion detection performance.

Secondly, the low-level feature is superior to the mid-level features. The mid-level feature is frequently used in scene classification problem because of its good ability to distinguish different scenes [38]. However, every scene has different congestion level which will confuse the detection of congestion level. In contrast, the low-level texture features can better reflect congestion level since most congested scenes have dense texture.

Lastly, The most popular regression method is Linear Regression (LR). However, Kernel Regression (KR) is shown to be superior than LR in the experiment. That indicates that the non-linear classifier is superior to linear classifier for congestion level regression. It is unsurprising since the cross scenes congestion level detection is a non-linear regression problem.

5.4. The Effectiveness of Locality Constraint Metric Learning

To confirm the effectiveness of Locality constraint Metric Learning, the Metric Learning for Kernel Regression is included as the comparison. A visualization of the regression results is shown in Figure 10.

As shown in Figure 10, the performance of Linear Regression is the worst since the congestion regression is a non-linear problem. The performance of Metric Learning for Kernel Regression is better. However, the influence among different scenes makes the prediction a hard work. The proposed Locality Constraint Metric Learning outperforms other methods since the effect among different scenes is reduced.

To further confirm that the locality constraint metric learning can efficiently reduce the influence of different scenes. An image retrieval experiment is conducted. In this experiment, 9 images from different scenes are randomly selected as the queries. 3 nearest neighbors of these queries are selected from the whole training set (containing 20 different scenes) with the proposed LCML. The result is shown in Figure 9.

As shown in Figure 9, most of the neighbors and queries are from the same scene and similar congestion level except for 3 failures which come from complex scenes that have no obviously road boundary. If all the training samples are included to predict the congestion level, the different scenes will influent each other, which decreases the detection performance. Only the neighbors which have same scene and similar congestion level are considered in LCML. Thus, the influence of different scenes is reduced and the performance of congestion level detection is increased with LCML.

6. Conclusion

To remedy the congestion detection problem, a locality constraint metric learning is proposed to reduce the influence among different scenes. A unified

Figure 9: Most of the neighbors and queries are from the same scene and similar congestion level with LCML. Only the neighbors which have same scene and similar congestion level are considered in LCML. Thus, the influence of different scenes is reduced and the performance of cross scene congestion level detection is improved. In this figure, the first column is queries and the rest are nearest neighbors. The images in red boxes are failures.

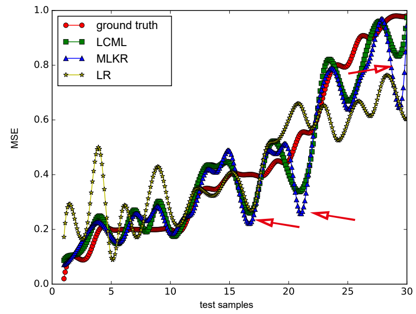

Figure 10: The visualization of the regression results. The proposed method outperforms MLKR, as the red arrows point.

and accurate definition of congestion is first proposed to better describe the traffic congestion level. Based on the definition, a dataset consists of 20 different scenes is constructed to serve as a platform for congestion analysis. To solve this problem, the low-level texture feature and kernel regression which outperform mid-level feature and linear regression are included as the feature and classifier. Since the influence of different scenes makes the detection of traffic congestion a difficult task, a locality constraint metric learning is proposed to reduce such an influence. The extensive experiments confirms the effectiveness of the proposed method.

In the future, some density based features shall be exploited to better represent the congestion level. Besides, a hierarchical model shall be exploited as well since the congestion level is a high level semantic conception.

References

[1] F. L. Hall, Traffic stream characteristics, Traffic Flow Theory. US Federal Highway Administration.

[2] L. Li, L. Chen, X. Huang, J. Huang, A traffic congestion estimation approach from video using time-spatial imagery, in: International Conference on Intelligent Networks and Intelligent Systems, 2008, pp. 465–469.

[3] F. Porikli, X. Li, Traffic congestion estimation using hmm models without vehicle tracking, in: IEEE Intelligent Vehicles Symposium, 2004, pp. 188–193.

[4] J. Lu, G. Wang, P. Moulin, Localized multifeature metric learning for image-set-based face recognition, IEEE Transactions on Circuits and Systems for Video Technology 26 (3) (2016) 529–540.

[5] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, J. Zhou, Neighborhood repulsed metric learning for kinship verification, IEEE Transactions on Pattern Analysis and Machine intelligence.

[6] S. Hu, J. Wu, L. Xu, Real-time traffic congestion detection based on video analysis, Journal of Information and Computational Science 9 (10) (2012) 2907–2914.

[7] C. Zhan, X. Duan, S. Xu, Z. Song, M. Luo, An improved moving object detection algorithm based on frame difference and edge detection, in: International Conference on Image and Graphics, 2007, pp. 519–523.

[8] B. K. Horn, B. G. Schunck, Determining optical flow, Artificial intelligence 17 (1-3) (1981) 185–203.

[9] A. Sobral, L. Oliveira, L. Schnitman, F. D. Souza, Highway traffic congestion classification using holistic properties, in: International Conference on Signal Processing, Pattern Recognition and Applications, 2013.

[10] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1981, pp. 674–679.

[11] K. G. Derpanis, R. P. Wildes, Classification of traffic video based on a spatiotemporal orientation analysis, in: IEEE Workshop on Applications of Computer Vision, 2011, pp. 606–613.

[12] A. Riaz, S. A. Khan, Traffic congestion classification using motion vector statistical features, in: International Conference on Machine Vision, 2013, pp. 90671A–90671A.

[13] E. Dallalzadeh, D. Guru, B. Harish, Symbolic classification of traffic video shots, in: Advances in Computational Science, Engineering and Information Technology, 2013, pp. 11–22.

[14] Y. He, Y. Mao, W. Chen, Y. Chen, Nonlinear metric learning with kernel density estimation, IEEE Transactions on Knowledge and Data Engineering 27 (6) (2015) 1602–1614.

[15] A. Globerson, S. T. Roweis, Metric learning by collapsing classes, in: Advances in neural information processing systems, 2005, pp. 451–458.

[16] Z. Huang, R. Wang, S. Shan, X. Li, X. Chen, Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 720–729.

[17] H. Chang, D.-Y. Yeung, Kernel-based distance metric learning for content-based image retrieval, Image and Vision Computing 25 (5) (2007) 695–703.

[18] W. Li, Y. Wu, J. Li, Re-identification by neighborhood structure metric learning, Pattern Recognition 61 (2017) 327–338.

[19] Z. Huang, R. Wang, S. Shan, X. Chen, Face recognition on large-scale video in the wild with hybrid euclidean-and-riemannian metric learning, Pattern Recognition 48 (10) (2015) 3113–3124.

[20] Z. Huang, R. Wang, S. Shan, X. Chen, Projection metric learning on grassmann manifold with application to video based face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 140–149.

[21] K. Q. Weinberger, J. Blitzer, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Advances in Neural Information Processing Systems, 2005, pp. 1473–1480.

[22] K. Q. Weinberger, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research 10 (2009) 207–244.

[23] J. V. Davis, B. Kulis, P. Jain, S. Sra, I. S. Dhillon, Information-theoretic metric learning, in: Proceedings of the International Conference on Machine Learning, 2007, pp. 209–216.

[24] J. Goldberger, S. T. Roweis, G. E. Hinton, R. Salakhutdinov, Neighbourhood components analysis, in: Advances in Neural Information Processing Systems, 2004, pp. 513–520.

[25] S. C. H. Hoi, W. Liu, M. R. Lyu, W. Ma, Learning distance metrics with contextual constraints for image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2072–2078.

[26] K. Q. Weinberger, G. Tesauro, Metric learning for kernel regression, in: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, 2007, pp. 612–619.

[27] R. Huang, S. Sun, Kernel regression with sparse metric learning, Journal of Intelligent and Fuzzy Systems 24 (4) (2013) 775–787.

[28] B. Xiao, X. Yang, H. Zha, Y. Xu, T. S. Huang, Metric learning for regression problems and human age estimation, in: Advances in Multimedia Information Processing, 2009, pp. 88–99.

[29] J. Hu, J. Lu, Y.-P. Tan, Discriminative deep metric learning for face verification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1875–1882.

[30] Z. Li, J. Tang, Weakly supervised deep metric learning for community-contributed image retrieval, IEEE Transactions on Multimedia 17 (11) (2015) 1989–1999.

[31] H. O. Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, CoRR abs/1511.06452.

[32] E. Y. Liu, Z. Guo, X. Zhang, V. Jojic, W. Wang, Metric learning from relative comparisons by minimizing squared residual, in: IEEE International Conference on Data Mining, 2012, pp. 978–983.

[33] H. Takeda, S. Farsiu, P. Milanfar, Kernel regression for image processing and reconstruction, IEEE Transactions on image processing 16 (2) (2007) 349–366.

[34] Y. Yuan, J. Wan, Q. Wang, Congested scene classification via efficient unsupervised feature learning and density estimation, Pattern Recognition 56 (2016) 159–169.

[35] J. A. Hartigan, M. A. Wong, Algorithm as 136: A k-means clustering algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics) 28 (1) (1979) 100–108.

[36] A. Coates, A. Y. Ng, Learning feature representations with k-means, in: Neural Networks: Tricks of the Trade - Second Edition, 2012, pp. 561–580.

[37] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 1794–1801.

[38] F. Li, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 524–531.