Pattern Recognition Coursework 2

Pu Yang
CID:01082988
Login: py715

Zhewei Zhang
CID:01092930
Login: zz8515

1. Introduction

In this coursework, we are solving the person re-identification problem using distance metric learning. The idea of person re-identification is to retrieve images containing the same person from gallery to the query set[10]. Generally, the metric learning algorithms include unsupervised learning such as PCA, KPCA, and supervised learning like Large Margin Nearest Neighbor (LMNN), Neighborhood Components Analysis (NCA) and Local Fisher Discriminant Analysis (LFDA). Both methods can transform the dataset to a new space, which aims to bring similar features closer; following by K-Nearest Neighbour (KNN), Nearest Neighbour (NN) or Kmeans algorithms we can find the accuracy of the re-identification algorithm. In this coursework, we implement several metric learning algorithms and they will be explained in the following sections. The source code can be found: ***HERE***.

2. Formulation

Person re-identification is the process to re-identify the person from one given image/video, given that the person appears from another camera with non-overlapping fields of views [3]. The process flowchart is shown in Figure 1. The varieties of people's action, object occlusion and the illumination condition make the re-identification process become more difficult. The problem is to find a robust way to extract the features to increase the accuracy of the re-identification process. As aforementioned, the images from two cameras have large variance within the classes and high similarity between the classes [2]. Hence, Euclidean distance metric might not be powerful enough for this problem. So we are going to explore several different metric learning algorithms in order to improve the re-identification accuracy.

The dataset we used is CUHK03, which contains 14096 images of label $l = 1467$ pedestrians, each image has dimension $d \in \mathbb{R}^{2048}$. The image has been divided into three fields, including training, query and gallery. The training data $X_{train} \in \mathbb{R}^{n \times 2048}$, where n is the number of training set and n = 7368. Each query image will be used as input in order to look for the nearest neighbours in gallery set, the

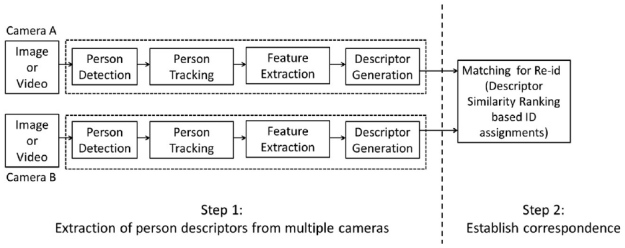

Figure 1: Re-Identification Process[11]

gallery data $X_{gallery} \in \mathbb{R}^{n \times 2048}$, where n is the number of gallery set and n = 5328. Query is the testing set that for each data in query set, $X_{query} \in \mathbb{R}^{n \times 2048}$, where n is the number of query set and n = 1400.

As aforementioned, we want to find a robust way to extract the features. In this case, using metric learning would help. Metric Learning methods is used for reducing dimensions or grouping features in order to transform the data points from old feature space to a new feature space that points with similarity will be placed closer, and vice versa [8]. The Distance metric learning methods can be formulated as following[13]:

$$\begin{aligned} \mathcal{L} = \min_{A} & \sum_{(x_i,x_j)\in\mathcal{S}} d(x_i,x_j)_A^2 \\ s.t. & \sum_{(x_i,x_j)\in\mathcal{D}} d(x_i,x_j)_A \geq 1, \\ & A \succeq 0. \end{aligned} \tag{1}$$

where the distance metric $d(x, y)$ is given that:

$$d(x,y) = d_A(x,y) = ||x-y||_A = \sqrt{(x-y)^T A(x-y)} \tag{2}$$

where $x_i, x_j \in X_{train}$, A is the covariance/similarity between the input x_i, x_j, $\mathcal{S}$ is the subspace containing similar points, and $\mathcal{D}$ is the subspace that containing di-similar points. By setting A = I, the distance metric will return Euclidean distance and if set A to be diagonal, the return distance will be calculated in Mahalanobis. By using such ideas from Equation 1, we can transform the data into a new

feature space in different distance measured scale.

We implemented a baseline method and several metric learning algorithms as an improvement of the baseline. We perform several comparisons using supervised and unsupervised metric learning algorithms including KPCA, LDA, Large Margin Nearest Neighbor (LMNN), Neighborhood Components Analysis (NCA), Local Fisher Discriminant Analysis (LFDA) and Metric Learning for Kernel Regression (MLKR). The details of the algorithm will be discussed in Section 3, 4 and 5.

3. Baseline Approach

In this section, we perform a baseline approach using K-Nearest Neighbour (KNN) and KMeans to perform the neighbour search. In order to evaluate the performance of the neighbour search, we use mean Average Precision (mAP) and Cumulative Matching Characteristic (CMC) to measure the metric learning performance. The original gallery data distribution plot is given:

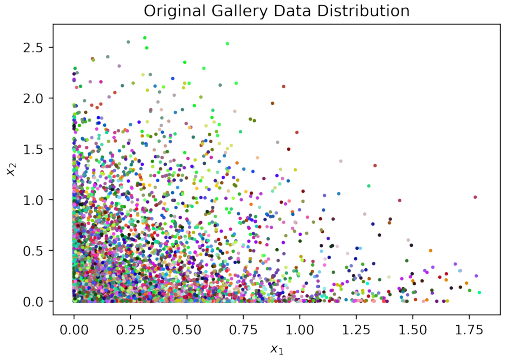

Figure 2: Gallery Distribution

3.1. Formulation

3.1.1 K-Nearest Neighbour

Regarding to distance metric, we aim to find N nearest neighbours of each query data, where N is our rank.
nn_k denotes our kth nearest neighbour, and x_i denotes the i-th data in gallery after deletion of same person's photos with same camid. m is the size of this gallery after the deletion, x is testing data from query, we have

$$\begin{aligned} nn_1(x) &= argmin_{i\in[M]}||x - x_i||^2 \\ nn_2(x) &= argmin_{i\in[M]\setminus\{nn_1(x)\}}||x - x_i||^2 \\ nn_3(x) &= argmin_{i\in[M]\setminus\{nn_1(x),nn_2(x)\}}||x - x_i||^2 \end{aligned} \tag{3}$$

where "\" in Equation 3 denotes 'remove elements from the set'.

and hence

$$knn(x) = \{nn_1(x), ..., nn_k(x)\} \tag{4}$$

The accuracy can be calculated as

$$\frac{1}{m}\sum_{j=1}^{m} I(\nexists y_{ij} = y_j),\ i \in knn(x_j) \tag{5}$$

where $k = N$.

3.1.2 KMeans

With this method, we use k-means algorithm to split the gallery set into k groups, then we use this to predict the group that each query data should be in. We are then looking at if we can find a sample with different camid but same label in the same group with the query data.
Accuracy is calculated as

$$\frac{1}{m}\sum_{j=1}^{m} I(\nexists y_{ij} = y_j),\ ij \in S(x_j) \tag{6}$$

where $S(x_j)$ denotes the class that x_j belongs to in the prediction.

3.1.3 Rank

Rank-k is the percentage of query images that have same label with the gallery image, where k means k-nearest neighbours[8].

3.1.4 Cumulative Matching Characteristic (CMC)

CMC is one of the metrics evaluation methods in person re-identification problem. The top k-ranked accuracy is given[1]:

$$\text{Accuracy}_k \begin{cases} 1 & \text{if @rank-k gallery label contains query label} \\ 0 & \text{otherwise} \end{cases}$$

3.1.5 mean Average Pricision (mAP)

mAP is another metric evaluation methods that can show us that how accurate that the prediction is and how good the model is to find all the possible positive cases in the top k ranked in KNN[5][7][9].We define precision as:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ &= \frac{TP}{\text{Total Positive Result}} \end{aligned} \tag{7}$$

where TP means True Positive prediction when actual label equals to the predicted label, and FP is False Positive prediction when actual label is not equal to the predicted label.

3.2. Performance

During the experiment, we perform the baseline using KNN, and KMeans is also used as an additional baseline

methods. The performance of both methods using different evaluation techniques are given:

Baseline Performance Accuracy (%)			
Score\KNN Metric	Euclidean	Chebyshev	Manhattan
rank 1	47.000	34.286	47.214
rank 5	66.857	52.286	66.143
rank 10	74.929	61.857	75.071
mAP	0.499	0.499	0.499

Table 1: Baseline Performance Accuracy

The default distance metric will be used in KNN is Eudlidean distance. From Table 1, we can see that the highest accuracy is achieved using Manhattan distance in KNN, and Chebyshev is the worst case. Refer to the "Trainin Instruction", the reference accuracy measurement is given: top1: 0.469286, top5: 0.668571, top10: 0.750000, mAP: 0.432923. We can say that our baseline method using Manhattan distance is the best among other two distance metrics and it works better than the reference values. Manhattan distance in this case is more preferable than Euclidean given that the input data has high dimension[4]. The reason is that Manhattan is L1 distance measure, but Euclidean is in L2. The difference between L1 and L2 is that L1 will have larger distance output than L1 if the difference between the dataset in every direction is bigger; in other words, L1 regards difference in more dimension as larger difference than larger difference in one dimension. Our data input has high difference in every dimension instead of in some direction. The accuracy plot using Manhattan distance in KNN is given:

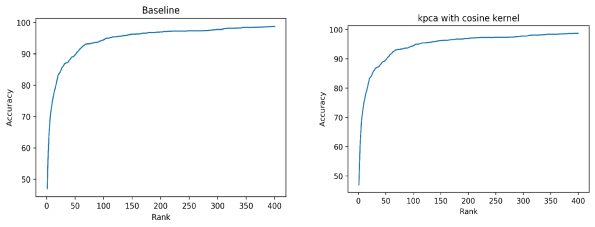

Figure 3: Baseline Accuracy using Manhattan Distance and KPCA accuracy

4. Improved Approach and Evaluation

Apart from baseline method, we proposed several different metric learning methods in order to find a hypothesis that can transform the data from old space to a new space and bring similar data points together, at the same time moving maximum the distances between the dissimilar points classes. As aforementioned X_{train} is the training set, we use X_{train} as the input of the metric learning algorithms. But before implemented the algorithm, we need to

KPCA Performance Accuracy (%)			
Score\Dist Metric	Euclidean	Manhattan	Chebyshev
rank 1	47.571	45.071	39.429
rank 5	67.000	65.786	61.214
rank 10	75.071	73.643	69.071
mAP	0.514	0.492	0.439

Table 2: KPCA Performance Accuracy

divide X_{train} into training and validation set. In practice, the validation set X_{val} is created using 100 random pedestrians from from X_{train}, the 100 randomly selected pedestrians with label l_{val_i} will be removed from the training set in order to access the generalization of the approach.

4.1. Kernel PCA

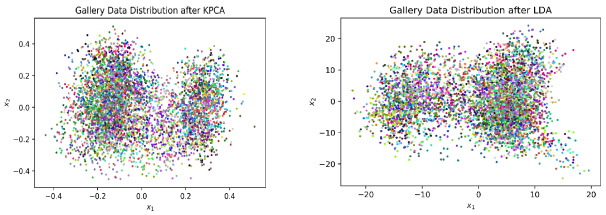

Figure 4: Gallery Distribution after KPCA and LDA

Kernel PCA is used to improve the accuracy and it can transform our data to a higher dimension by kernels, then performing PCA to maximize the total distance between each pair of the data. In this way the component analysis can deal better with the non-linearity data.

Choosing a right kernel is challenging. At first we used polynomial kernel, which gives accuracy worse than baseline. We find that Radial basis function is good to use when we do not have prior knowledge on data, so we tried RBF kernel and the accuracy is similar to baseline (better than polynomial kernel). Then we tried cosine kernel and it gives a improved accuracy, comparing to baseline.

The feature data for photos may tend to be periodic, in each period the pattern is similar, so a cosine kernel fits the feature space better. We can say cosine kernel is a correct prior while polynomial kernel is not a correct prior in this case. Figure 4(left) and Figure 4(Right) shows us the distribution of gallery data after the transformation. Compare to Figure 2, we can see that the data points are mapped to new space with more separation and higher between class variance, which also prove that the requirement and the constrain in Equation 1 have been satisfied.

Performance Accuracy (%)				
Score\Metric	LFDA	LMNN	MLKR	NCA
rank 1	42.429	45.571	47.000	44.643
rank 5	61.286	66.000	66.857	66.000
rank 10	70.929	74.000	74.929	74.000
mAP	0.457	0.484	0.499	0.481

Table 3: Supervised Metric Performance Accuracy

LDA Performance Accuracy (%)			
Score\Metric	Euclidean	Manhattan	Chebyshev
rank 1	42.143	41.571	33.000
rank 5	62.143	60.429	52.714
rank 10	71.714	69.214	60.857
mAP	0.469	0.456	0.377

Table 4: LDA Performance Accuracy

4.2. LDA

We also tried LDA. LDA maximize the distance between different classes while minimize the distance between points in the same class. The accuracy is worse than the baseline. Without a kernel, LDA is done in low dimension, resulting in less information kept. The advantage of LDA towards baseline is the speed, since it has less dimension, the KNN runs faster than baseline.

4.3. Supervised Metric Learning

In this section, we implemented the Metric Learning algorithms using Supervised Learning by providing both image data and the person label to the algorithm in order to "guide" the machine learning algorithms. We implemented several supervised algorithms including Large Margin Nearest Neighbor (LMNN), Neighborhood Components Analysis (NCA) and Local Fisher Discriminant Analysis (LFDA) and Metric Learning for Kernel Regression (MLKR). Table 3 shows the performance of those algorithms.

Note, LMNN, NCA, LFDA and MLKR are the algorithms that using Mahalanobis distance matrix. Mahalanobis is one of the methods that using the class-specific covariance matrices to calculate the distances, Mahalanobis distance not only describe the difference two data points, it also includes the relationship of the features, where the relationship is calculated from the total samples. A pair of data will have different Mahalanobis distance if they are from different data sets. This is better than Euclidean if we are looking for centralized and normalized distance.

4.3.1 LMNN

LMNN is a method try to make sure the k nearest neighbours of each sample is in the same group with it, and all data from different group at least have a smallest distance d, from this data point. LMNN minimize the number of impostor. For a data point, a impostor is its k nearest neighbours in different class from the point.

4.3.2 MLKR

MLKR is the metric learning algorithm that using kernel regression, where the kernel function $k(x_i, x_j) = k_D(D_\theta(x_i, x_j))$. The algorithm will differentiate the loss function $\mathcal{L} = \sum_i (y_i - \bar{y})$ to perform gradient descent to update θ, where $\bar{y}$ is the weighted average of the training set. [12]

4.3.3 NCA

Neighbourhood Component Analysis use Mahalanobis distance metrics as learning method to transform data to a new feature space as we mentioned before. As we do not know the whole data set, in this case, NCA attempts to optimize the leave-one-out performance on the training set. The probability of each point i correctly selecting another point j as its stochastic neighbour is defined as p_{ij} [6]:

$$p_{ij} = \frac{exp{-}||Ax_i - Ax_j||^2}{\sum_{k \neq i} exp{-}||Ax_i - Ax_j||^2}, p_{ii} = 0 \tag{8}$$

and the objective function is given:

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum p_i \tag{9}$$

where C_i is the i-th class. Maximized the objective function can hence maximized the probability of correctly classified point i.

5. Evaluation and Conclusion

To sum up, we perform several distance metric learning methods in both supervised and unsupervised case. The best algorithm in this person re-identification problem is Kernel PCA using cosine as kernel, the reason has been discussed in Section 4.1 as the feature data for photosmay tend to be periodic, in each period the pattern is similar,so a cosine kernel fits the feature space better in this case. The worst case scenario is using LDA, this is because we did not perform Kernel LDA. Furthermore, although KPCA is the best case in our experiment, we still believe that there will be a better improvement using other algorithms like Convolution Neural Network, Neural Network or Kernel LDA.

6. Appendix

User Guide:

- Download the zip file or git clone the repository from ***Our GitHub : https://github.com/xxyypp/Pattern-Recognition-CW2***
- Our source codes are in both Jupyter notebook (*.ipytb) and Python (*.py) format
- File name indicates which methods that we used, for example: "baseline.ipytb" and "baseline.py" are the source code for the baseline method; `"kpca_rbf.ipytb" and "kpca_rbf.py"` are the KPCA methods using RBF kernel.
- If open Jupyter Notebook file, simply click run in the Notebook.
  If use .py file, compile and run the file in the compiler/editor.
- For a better visualization, we recommend using Jupyter Notebook to open the source code, since the python file is the converted version of the Jupyter Notebook file.

References

[1] Evaluation metrics.

[2] J.-P. Ainam, K. Qin, and G. Liu. Self attention grid for person re-identification. /09/23 2018.

[3] A. Bedagkar-Gala and S. K. Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4):270–286, April 1, 2014.

[4] A. H. Charu C. Aggarwal and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space.

[5] J. Hui. map (mean average precision) for object detection, -03-07T00:05:23.232Z 2018.

[6] G. H. R. S. Jacob Goldberger, Sam Roweis. Neighbourhood components analysis.

[7] W. Koehrsen. Beyond accuracy: Precision and recall.

[8] M. Krystian. *Pattern Recognition Lecture Notes*. 2018.

[9] M. Malaeb. Recall and precision at k for recommender systems, -08-13T20:07:08.213Z 2017.

[10] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval, Mar 16, 2017.

[11] N. Viswarupan. A practical guide to person re-identification using alignedreid, -06-25T06:30:35.322Z 2018.

[12] K. Q. Weinberger and G. Tesauro. Metric learning for kernel regression.

[13] E. P. Xing, A. Y. Ng, M. I. Jordan, S. R. U. of California, B. Berkeley, C. . epxing, ang, jordan, and russell @cs.berkeley.edu. Distance metric learning, with application to clustering with side-information, 2003.