EE4-68 Pattern Recognition Coursework 2

Representation and Distance Metric Learning

Zihan Liu
Login: ZL6114
CID; 00955065
Email: zl6114@ic.ac.uk

Aufar Laksana
Login: APL115
CID; 01093575
Email: apl115@ic.ac.uk

Contents

1. Summary

The purpose of this project is to optimize the performance of a person re-identification problem. The goal is to match a pedestrian with its later images captured from different cameras. The studied data set (CUHK03) has 14096 samples and 1467 identities in total. This project focuses on the geometrical transformation and similarity estimation of the re-id problem. Different geometrical transformations are explained in this report. For similarity estimation; Nearest Neighbours and J-mean clustering methods are used, with both Euclidian and Mahalanobis distance. A few distance metrics approaches are detailed in this report, the top method being Kernel PCA with Cosine Kernel with 200 components.

2. Problem Introduction

The data set is split into training data and test data, with a goal of extracting a pattern $\mathbf{W}$ from the training data. The transformation is used to improve the future prediction by projecting the test data into the new space via the equation $\mathbf{Y} = \mathbf{W}\mathbf{X}_{test}$.

The training data $\mathbf{X}_{train} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{X}_{train} \in \mathbb{R}^{n \times F}$, the training label $\mathbf{Y} = \{\mathbf{y}_{cam_id}, \mathbf{y}_{pid}\} \in \mathbb{R}^{n \times 2}$ where $n = 7368$ and $F = 2048$, a smaller set is also extracted from the training set to perform the validation.

The test data consists of two data sets, The Query set, $\mathbf{X}_q \in \mathbb{R}^{1400 \times F}$,$\mathbf{Y}_q \in \mathbb{R}^{1400 \times 2}$, and the Gallery set $\mathbf{X}_g \in \mathbb{R}^{5328 \times F}$,$\mathbf{Y}_q \in \mathbb{R}^{1400 \times 2}$. The goal here is to use an unsupervised method to predict the queried person in the gallery set by retrieving the closest gallery entries for each query, and maximizing its accuracy. The test data is simulated as the future data in this problem, and the label for the test data is only used in performance calculation.

2.1. Mathmatical Formulation

The optimization problem is trying to maximize the accuracy between the target query q and the number of times a person is correctly identified in the ranklist $\mathcal{L}(q, \mathbf{P}_q) = \{\rho_1, ..., \rho_n\}$, where $\mathbf{P}_q \in \mathbb{R}^{(n-C) \times F}$, $n = 7368$ and $F = 2048$. $\mathbf{P}_q$ is a reduced set of the gallery data (An image containing the same identity and from the same camera angle is deleted). The goal can be mathematically formulated as below:

$$min \sum_{i=1}^{n} Y_{\mathcal{L}(q,\mathbf{P}_q)_i} - Y_q, \forall q \in \mathbf{Q} \quad (1)$$

$$s.t.\ \mathcal{L}(q, \mathbf{P}_q) = min \Sigma_{i=1}^{rank} \Sigma_{j=1}^{n} d(\phi(q), \phi(\rho_\mathbf{j}))_i \quad (2)$$

$$\mathbf{Q} = \Phi(\mathbf{X}_q), \mathbf{Q} = \{q_1, \ldots, q_n\} \quad (3)$$

$$\mathbf{P} = \Phi(\mathbf{X}_p), \mathbf{P} = \{q_1, \ldots, q_n\} \quad (4)$$

$\Phi(X)$ is the geometrical transformation techniques corresponding to the learned distance metric; used to improve the accuracy of the prediction, as detailed in Section 4

3. Baseline Learning

3.1. K-Nearest Neighbour Method

The KNN method is used to calculate the ranklist of each query. The KNN formulation can be characterized as below, note that $\mathbf{X}_{p,q_i}$ is the reduced set of the gallery data, q_i:

$$\sum_{i=1}^{1400} \mathcal{L}(\mathbf{x}_{q_i}, \mathbf{X}_{p,q_i}) = min \sum_{j=1}^{rank} \sum_{k=1}^{n_i} d(\mathbf{x}_q, \mathbf{x}_{p_k,q_i})_j \quad (5)$$

$$where\ \mathbf{X}_q = \{\mathbf{x}_1, \ldots, \mathbf{x}_{1400}\}, \mathbf{X}_p = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad (6)$$

Intuitively, the goal is to find the N nearest neighbours for each query in its reduced gallery. The precision is then calculated by comparing the labels of the rank-list and the label of the query. Table 1 shows an example of the prediction and the rank-list for 10 neighbours. In the table, 1 denotes a correct prediction, and 0 are incorrect predictions.

Query	Label	Nearest Neighbours Predictions									
570	575	1	1	0	0	0	0	1	0	0	0
70	86	1	1	1	1	0	0	0	0	0	0
690	716	1	1	1	1	0	0	0	0	0	0

Table 1: Examples of Euclidean KNN at Rank10

Euclidean Distance The most commonly used distance calculated by:

$$d(\mathbf{x}_q, \mathbf{x}_p) = \sqrt{(x_{q_1} - x_{p_1})^2 + \cdots + (x_{q_n} - x_{p_n})^2} \quad (7)$$

Manhattan Distance This distance measures the sum of the absolute differences of their Cartesian coordinates, which can be formulated as:

$$d(\mathbf{x}_q, \mathbf{x}_p) = \Sigma_{i=1}^{n} |x_{q_i} - x_{p_i}| \quad (8)$$

Covariance Mahalanobis Distance This is a distance measurement method which measures the standard deviation distance between the two targets, Mahalanobis distance can be measured as:

$$d(\mathbf{x}_q, \mathbf{x}_p) = \sqrt{(\mathbf{x}_q - \mathbf{x}_p)^T \mathbf{S}_{\mathbf{x_q}, \mathbf{x_p}} (\mathbf{x}_q - \mathbf{x}_p)} \quad (9)$$

$$where \ \ \mathbf{S}_{\mathbf{x_q}, \mathbf{x_p}} = \mathbb{E}[(\mathbf{x}_q - \mathbb{E}[\mathbf{x}_q])(\mathbf{x}_p - \mathbb{E}[\mathbf{x}_p])] \quad (10)$$

3.1.1 Score at each rank

If there is at least one correct prediction in the ranklist, the score is registered, below shows the score at rank 1,5,10 for the baseline approach with Euclidean distance, the score increases as the number of rank increase, due to the increased number of neighbours. The complete graph of all distances is shown in Figure 7 in the appendix, showing Euclidean distance to have the exact same score as Manhattan distance.

Rank	1	5	10
Score(Euclidean)	47.2%	66.1%	75.1%
Score(Manhattan)	47.2%	66.1%	75.1%
Score(Mahalonobis)	46.6%	65.6%	74.4%

Table 2: Score at different rank for euclidean distance

3.1.2 Mean average precision

Mean average precision calculates the average of the maximum precisions at different recall levels. Recall is the percentage of correct predictions at a given point compared to the total number of correct predictions. In this question an interpolation step is performed. The full equation to calculate mAP is listed below, where the precision with a retrieval cutoff of k images:

$$\sum_{k=1}^{N} max_{\bar{k} \geq k} P(\bar{k}) \Delta_r(k) \quad (11)$$

The interpolated mAP for this problem is 51.7%. Figure 8 in the appendix shows the difference between the interpolated and normal average precision.

3.2. K-mean

Another baseline approach used is the K-mean method. The goal of the method is to partition the n observations into the number of unique sets that exist in the set, by minimizing the variance of each cluster. The center of each unique class is randomly assigned at the beginning, then the nearest observation to the randomly assigned center is assigned to that cluster. After all observations are assigned, the center is then moved to the mean of each observation. The later iterations with the same algorithm are performed until convergence. The formulation can be expressed as below:

$$\underset{\mathbf{s}}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{1}{2|S_i|} \sum_{\mathbf{x}, \mathbf{y} \in S_i} ||\mathbf{x} - \mathbf{y}||^2 \quad (12)$$

In this problem the unique data set has 700 components. After the cluster algorithm is completed, each query is then project into the new space. The cluster the query belongs to is calculated. The same person-id taken from the same camera is not considered as a previous observation. The baseline accuracy at random_state = 0 is 52.5%.

4. Improved Approach

The proposed improvement is to use distance metrics to extract a pattern from the training data, validate the result, and use the learned distance metrics to transform the gallery and query data. Ideally, this would improve the accuracy by increasing the between class distances, and reducing the distances between points of the same class.

4.1. Principal Component Analysis

Due to the large number of samples and features in the data set, the computational complexity of many of the methods increases exponentially. Thus, subspace learning method, for example, the Principal Component Analysis can be used to reduce the dimension of the data.

By projecting the data into a space spanned by the principal components, it is possible to reduce the number of the features required to represent the data, thereby reducing the complexity of the distance learning methods applied.

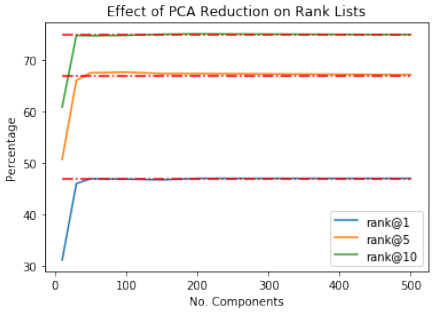

Figure 1: PCA Dimension Reduction on Rank List

Figure1 shows that the minimum number of components to achieve a similar retrieval score to the baseline is at $n = 30$,

with $rank@1 = 46.0\%, rank@5 = 66.071\%, rank@10 = 74.786\%$, which are slightly lower than the baseline retrieval errors.

4.2. Validation

A validation set is extracted from the training data to evaluate the performance of the trained geometric transformation. To create the validation set, all labels of the same person needed to be removed from the training set to the validation set. The methods to evaluate the validation set is the same method used in the baseline approach (KNN).
In this problem, 100 randomly selected identities are extracted from the training set to form the validation set.

4.3. Large Margin Nearest Neighbor

The objective of LMNN algorithm is to find a subspace transformation that minimizes the with-in class distance, whilst maximizing the between class distances by learning a Mahalanobis distance for the data. The distance can be expressed as:

$$D(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \mathbf{x}_j)^T M (\mathbf{x}_i \mathbf{x}_j) \tag{13}$$

where M is the Mahanobis distance learned from the training data, M can be denoted to $L^T L$, and L can be calculated via, the result performs poorly in the validation set, scores average -5% at rank 10 compares to baseline method.

4.4. Kernel Principal Component Analysis

From Section. 4.1, it was shown that normal Principal Component Analysis had a slight loss in retrieval scores when the data is projected to a space with a lower number of components. This is due to the selected principal components not being able to represent all the original information. However, this was done in order to decrease the complexity of subsequent distance learning methods.

4.4.1 Brief Explanation

Kernel PCA is often used to transform non-linear data into a feature space where the features allow for linear predictions. Essentially, KPCA transforms the original data into a new (possibly higher) dimension where a linear PCA is used to generate uncorrelated features.
The Gram Matrix is the inner product between two samples from the data $K_{ij} = \phi(x_i)^T \phi(x_j)$. In PCA, the principal components are computed through eigen decompositions. In Kernel PCA, the Gram Matrix allows for the calculation of the projection of the data onto the principle components, without explicitly computing them. This allows for the data to be transformed into the KPCA space.

4.4.2 Cosine Similarity Kernel

For this project, the Cosine Similarity kernel was used in Kernel PCA instead of the regular Euclidean Distance. Cosine similarity is a inner product space that returns the angle between the two non-zero vectors, and the smaller the angle mean higher similarity.The kernel can be defined as:

$$k(x, y) = \frac{\mathbf{x}\mathbf{y}^T}{\|\mathbf{x}\|\|\mathbf{y}\|} \tag{14}$$

Cosine Similarity can be considered as a normalized Euclidean Distance.The Cosine Similarity kernel is often used when the scale of the features varies across the data. The graph in Figure. 2 shows that there are several features that have a larger scale.

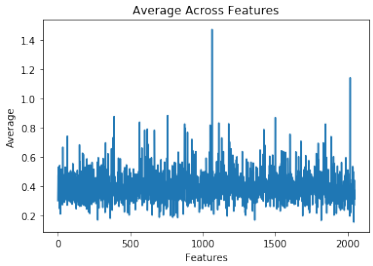

Figure 2: Average Feature Value

4.5. Neighbourhood Component Analysis

Neighborhood Component Analysis learns a Mahalanobis Distance (the matrix **A**) and used to transform the input data into a space where the average leave-one-out classification performance is maximized. The matrix can be found by defining a differentiable objective function and finding the optima using an iterative solver.

Brief Explanation The class label of a single data point can be predicted by the consensus of its k-nearest neighbours. However, after applying the transformation **A**, the nearest neighbours of the same data point may be completely different.
In NCA, for each transformed point, the entire transformed dataset is considered as its stochastic nearest neighbours. Essentially, the probability of correctly classifying a point p_i is the sum of the probabilities of correctly classifying its neighbours (p_{ij}) which are of the same class C_i.
The probability of each neighbour being of the same class as p_i is defined by:

$$p_{ij} = \frac{e^{-\|Ax_i - Ax_j\|^2}}{\sum_k e^{-\|Ax_i - Ax_k\|^2}} \; for\, i \neq j, \; 0\, otherwise$$

The objective function is then defined as:

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

Where the objective function is maximized so that the probability of correctly predicting the class of the leave-one-out

point is maximized. Under stochastic nearest neighbours, the consensus class of a point is determined by the average class of all its neighbours.

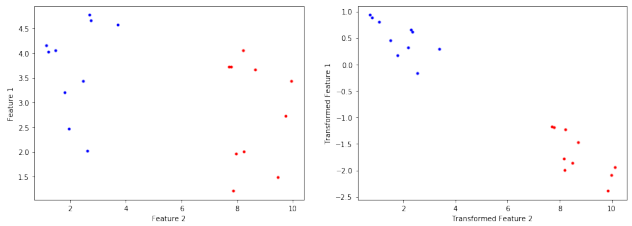

Figure 3: A toy example showing the NCA Transformation

Intuitively, the transformation matrix **A** transforms the input data into a space where points of similar classes are closer together, whilst points of different classes are further apart. This can be seen in Figure 3, where the transformation squeezes the same classes together.

4.6. KPCA + NCA

By first passing the data through a Cosine Similarity kernel in a KPCA, the dimensionality of the input data is reduced. This significantly speeds up the computation time of the NCA, which needs to iteratively calculate the gradient of the objective function in order to find the maxima.

5. Evaluation

In the Figure 4, the table shows that actually, KPCA with a cosine similarity kernel actually outperforms the baseline, NCA, and the KPCA + NCA combination.

Method	rank@1	rank@5	rank@10
Baseline	47.0	66.857	74.927
NCA	44.642	65.929	73.857
KPCA (200)	47.643	67.357	75.071
KPCA (1000)	47.357	67.142	75.071
KPCA (200) + NCA	45.429	66.214	75.357
KPCA (1000) + NCA	41.857	62.214	70.571

Figure 4: Comparison of Different Methods

As shown in Section 4.1, dimensionality reduction decreases the retrieval score below baseline. This can be seen in Figure 5. The rank@10 score for KPCA + NPCA is higher than that of the baseline. This shows that in larger neighbour sets, there are now potentially more neighbours with the correct class label.
The lack of a significant increase suggests that the metric learned by the NCA in the KPCA space did not significantly change the neighbours of each query point. The figure further shows that as the number of components used is increased, the retrieval scores decline for all ranks.

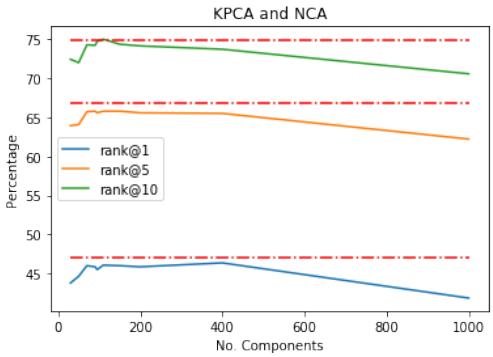

Figure 5: KPCA Feature Reduction & NCA

A possible reason why KPCA with 200 components outperforms the other methods proposed may be due to kernel used. By scaling the features, it is possible to lose information, and as such, reconstruction is impossible. However, since the goal of the task was to find a space which maximizes the number of correct nearest neighbours in the rank list, this was deemed an acceptable loss.
The Cosine Similarity can also be thought of as comparing the two feature vectors and seeing where both vectors overlap. The less overlap there is between the two vectors, the greater the cosine angle (the vectors are more dissimilar).

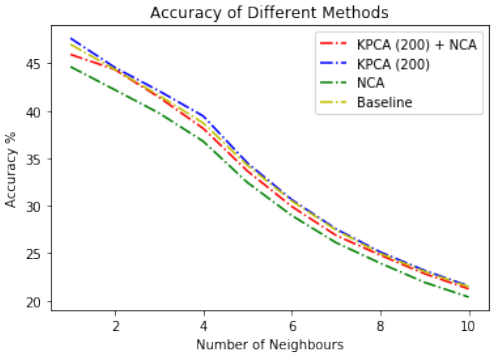

Figure 6: Accuracy of Different Methods

Conclusion The Figure. 6 shows that the accuracy of the methods decrease as the number of neighbours is increased. The baseline consistently outperforms the NCA method. KPCA+NCA and KPCA outperform the baseline at the lower ranks. Figure 9 in the Appendix shows the mAP scores of the different methods. It clearly shows that the KPCA with cosine kernel is the best performing method.

References

[1] Christopher M. Bishop., *"Pattern Recognition and Machine Learning"*, ISBN: 1493938436.

[2] Jacob Goldberger,Sam Roweis, Geoff Hinton, Ruslan Salakhutdinov *Neighbourhood Components Analysis*

[3] Kilian Q. Weinberger, Lawrence K. Saul *Distance Metric Learning for Large Margin Nearest Neighbor Classification*

[4] Aleksei Tiulpin, *"A Tutorial On Kernel Principal Component Analysis"*, `https://atiulpin.wordpress.com/2015/04/02/a-tutorial-on-kernel-principal-component-analysis/`

[5] Python, *"Scikit-learn Documentation"* `https://scikit-learn.org/stable/documentation.html`

[6] Python, *"The Metric Learn Project"* `https://pypi.org/project/metric-learn/`

[7] Jonathan Hui, *"mAP (mean Average Precision) for Object Detection"* `https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173`

A. Appendix

Source Code

`https://github.com/zl6114/pattern-recognition-18/tree/master/cw2`

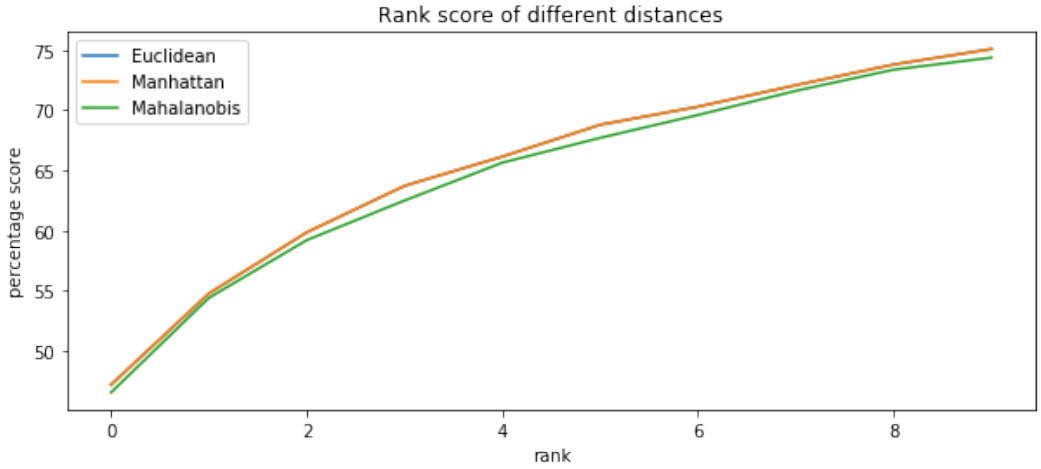

Figure 7: The score of the baseline apparches with different distance calculation

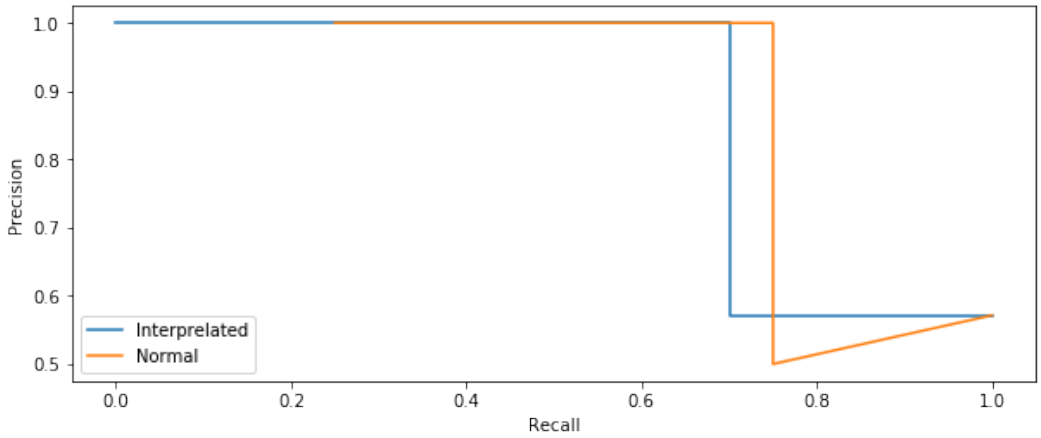

Figure 8: Interprelated vs Normal

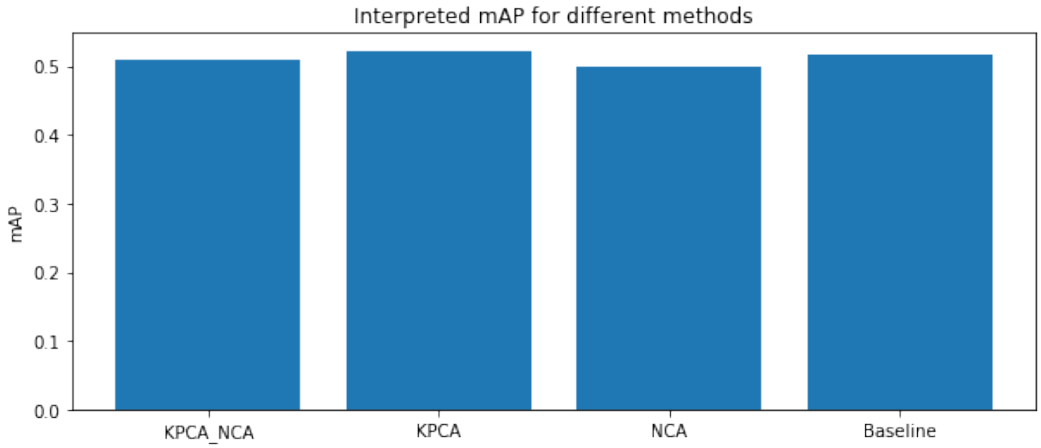

Figure 9: Interprelated mAP at different methods