EE4-68 Pattern Recognition Report

Kaiyue Sun
CID: 01197813
kaiyue.sun16@imperial.ac.uk

Husheng Deng
CID: 01178574
husheng.deng16@imperial.ac.uk

Abstract

We present an exploration of non-learned and learned distance metrics by considering the retrieval process involved in face recognition. It is a task of correctly identifying the the face images of same individuals. Our objective is to improve the baseline approach that performs K-Nearest Neighbour retrieval on the data representations of image pixel. We endeavour to find an optimal approach that minimises the retrieval error which is reflected by the performance scores (@Rank1, @Rank10, mAP)

1. Q1

1.1. The Baseline Approach

The face images consist of 520 2576-dimension vectors, with the first 320 as the training split, remaining 200 as the test split. The feature vectors are prepared using the original unmodified images (A.a) and using the normalised images to unit norm L_2 (A.b). The baseline approach performs k-nearest neighbors algorithm on the two types of feature vectors, and the performances are evaluated using standard metrics of Minkowski-Form Distances ($L_{1-5,inf}$), Cosine and Cross correlation Similarities. The Cosine distance normalises the feature vectors and measures the angle in between, so it is the same for the original and normalised vectors. Cross correlation takes the dot product of two vectors. This distance applied on the normalised vectors is identical to the Cosine distance. However, the cross correlation applied on unmodified positive vectors is largely affected by the magnitudes, its value is not accurately associated with the similarity. We cannot predict the similarity between two unnormalised vectors by using this distance. Figure 1, 2, and 3 report the performance scores Average Accuracy @rank-1 (Acc_1), Average Accuracy @rank-10 (Acc_{10}), and Mean Average Precision (mAP) of different metrics respectively for the test spilt. The horizontal axis represents the scores on A.b, the vertical axis is for A.a. The probability that at least one image from the same class exists in the top 10 closest images(Acc_{10}) is about 0.8 to 1.0 for most metrics except L_{inf}. The probability that the closet image belongs to the correct class (Acc_1) is much lower than Acc_{10} because it implements a much harder restriction and it is more spread out for different metrics, generally below 0.8. L_1 is the best, 0.755 for A.b, the higher the order is, the worse the score becomes. For the best two metrics, L_1 and L_2, normalised feature vectors result in higher accuracy and mAP than original vectors. Cosine is most similar to L_2. The detailed scores are shown in Table 1.

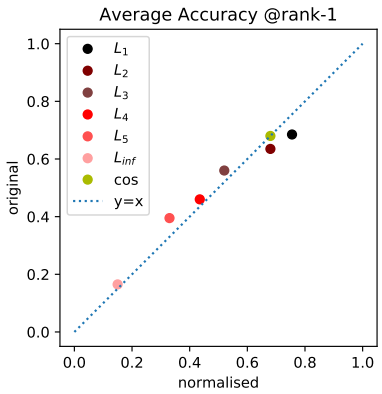

Figure 1: Average Acc_1 of Standard Distance Metrics

Table 1: Performance of metrics {normalised ; original}

Metrics	@rank1	@rank10	mAP
L_1	0.755 ; 0.685	0.955 ; 0.935	0.263 ; 0.242
L_2	0.680 ; 0.635	0.945 ; 0.945	0.230 ; 0.227
L_3	0.520 ; 0.560	0.910 ; 0.920	0.191 ; 0.199
L_4	0.435 ; 0.460	0.875 ; 0.890	0.162 ; 0.168
L_5	0.330 ; 0.395	0.825 ; 0.850	0.139 ; 0.146
L_{inf}	0.150 ; 0.165	0.565 ; 0.620	0.0618 ; 0.0675
Cosine	0.680	0.945	0.2299

1.2. Experiment 1

Histograms of original and normalised pixel intensities (C.a, C.b) are implemented as the feature representations.

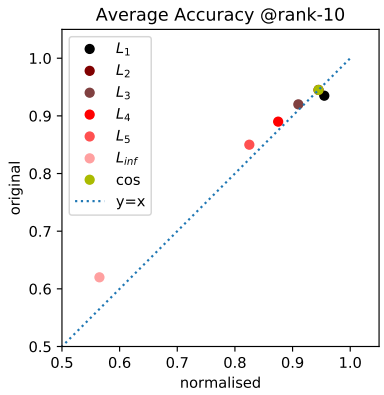

Figure 2: Average Acc_{10} of Standard Distance Metrics

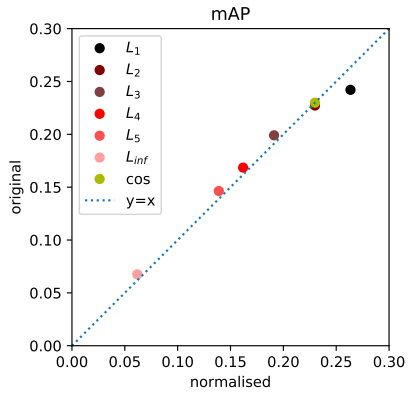

Figure 3: mAP of Standard Distance Metrics

The metrics of Euclidean, Intersection and Chi-square are used. The Intersection Distance is the same for the original and normalised histograms. By increasing the number of bins from 5 to 500 in a step of 1, the variation of the accuracies are plotted in Figure 5-13 in Appendix. The original and normalised histogram representations have very similar scores. The scores on original histograms of critical bin numbers ($k = \sqrt{n}$, $[\log_2 n] + 1$, $2n^{1/3}$) are summarised in Table 6 in Appendix. Intersection gives the highest scores. From the figures, Euclidean and Intersection reveal very similar trend and fluctuation. This is because measuring the common parts or different parts of histograms have the same effect. All cells are weighted equally. What is noticeable is that at about one tenth of the dimension, the Acc_1 stops falling and rises abruptly almost to the highest score. Chi-square shows less fluctuant accuracies. Since cells are weighted by frequency, both similarity and difference between large bins are emphasised, outliers have little effect. Therefore, Chi-square works better for reasonably small bin number and large size bins, which generalise the distribution. When bin number is too large, each bin does not contain sufficient pixels, so is like an outlier. Thus, Chi-square cannot catch the intrinsic distribution when size of bin is too small. When bin number is over 280, Acc_1 converges to 0.05, which is the average score, 10/200, with no meaningful clustering. But Acc_{10} converges to 0.1, which is much smaller than the 0.4 of the other two metrics. The decrease of the accuracy with generally with increasing bin number is due to less refined feature extraction with smaller bins.

1.3. Experiment 2

The standard Mahalanobis distance metric is implemented by calculating the covariance matrix on the training split of features A.a and A.b. By applying it to the 2576-dimension test split of features A.a and A.b respectively, the score is reported in the last row in Table 2. However, as the image covariance matrix is not full rank, it is possible to carry out dimensionality reduction by analysing the principal components of the image vector space, and use the projections of images in the lower-dimension space for more efficient distance metric computation.

Experiments have been carried out on the application of principal component analysis to project both the training and testing images to low-dimensional spaces before the computation of Mahalanobis distance for retrieval. From the result in table 3, as the number of principal components from the PCA model decreases, the retrieval performance generally increases, up to 32 principal components. This is because the less important components found by the PCA model may act as noise in the actual distance computation. However, the performance at 16-dimension PCA is worse than 32-dimension PCA, because the sacrifice of too many components have rendered the PCA model estimation less accurate. However, with suitable selection of the number of principal components, retrieval in the transformed space may be better than retrieval in the original image space.

Table 2: Performance of Mahalanobis distance {normalised ; original}

Dimension	@rank1	@rank10	mAP
16	0.595 ; 0.590	0.950 ; 0.930	0.221 ; 0.234
32	0.655 ; 0.685	0.950 ; 0.955	0.237 ; 0.244
64	0.640 ; 0.660	0.880 ; 0.895	0.209 ; 0.215
128	0.365 ; 0.360	0.635 ; 0.710	0.107 ; 0.103
256	0.125 ; 0.110	0.420 ; 0.510	0.053 ; 0.0473
2576	0.590 ; 0.635	0.875 ; 0.895	0.216 ; 0.221

1.4. Experiment 3

Firstly, PCA-LDA is performed on A.a and A.b with $M_{pca} = 50$, $M_{lda} = 19$. The performance is evaluated using L_1 and L_2. Then it is done on C.a and C.b with same

parameter values. Bin number = 51 when using Euclidean Distance to evaluate the performance, and bin number = 14 when using Chi-square Distance The scores are summarised in Table 3. By using PCA-LDA, different classes are separated, therefore, the scores of feature A are very high and almost 100%. However, by using the histograms as the feature vectors, the relative positions of pixels are discarded. PCA-LDA can hardly capture the characteristics of the face images. Thus the scores on features of C are comparable to statistical average value with no clustering.

Table 3: Performance with PCA-LDA {normalised ; original}

Metrics	@rank1	@rank10	mAP
$A(L_1)$	0.990 ; 0.980	1.000 ; 1.000	0.769 ; 0.758
$A(L_2)$	0.990 ; 0.995	1.000 ; 1.000	0.776 ; 0.774
C(Euc)	0.185 ; 0.175	0.505 ; 0.490	0.125 ; 0.124
C(Chi-sqr)	0.055 ; 0.070	0.395 ; 0.450	0.017 ; 0.029

1.5. Experiment 4.

1.5.1 Relevant Component Analysis

A typical distance is affected by all the variability that is maintained in the data representation. For a specific task, some of the variability is irrelevant. For example, when we want to retrieve a type of facial expression, the facial features are irrelevant. The removal of this variability won't deteriorate the results of clustering or retrieval. RCA is a linear method which compresses the data description along the dimensions of highest irrelevant variability. It transforms the data space by a matrix W which assigns high weights to relevant dimensions and low weights to irrelevant dimensions[2]. In face recognition, irrelevant variability indicates the variability within the same class, which can be characterised by the within class covariance matrix S_W. Its eigenvectors are ordered by eigenvalues, with each one representing the direction and amount of variation within the classes. By projecting the data onto $S_W^{-\frac{1}{2}}$, large within class variations are suppressed, for example, the lighting condition, hair style, glasses. While the small within class variations, the facial features, domains the new feature space. RCA also minimises inner class distances. Therefore, Mahalanobis distance can be performed on A.a and A.b of the test split by using S_W^{-1} of the training split. Each class can be partitioned into smaller chunks, S_W formed by the small chunks still maintains its property. The scores are summarised in Table 4

1.5.2 Neighbourhood Component Analysis

Neighbourhood component analysis(NCA) provides an alternative way of training the Mahalanobis distance metric by optimising the expectation of the number of points accurately classified[1]. With a softmax based differentiable objective function, gradient descent can be used to find local optimised solutions depending on the initial metric parameter.

Table 4: RCA Performance {normalised ; original}

Chunk size	@rank1	@rank10	mAP
10	0.450 ; 0.420	0.815 ; 0.850	0.155 ; 0.146
5	0.590 ; 0.555	0.870 ; 0.845	0.201 ; 0.178
2	0.660 ; 0.525	0.920 ; 0.875	0.217 ; 0.186

Due to the large pair-wise Euclidean distances involved in the soft-max computation, NCA learning algorithm exhibits considerable numerical instability when training with raw images that causes 64-bit floating point underflow with python when calculating exponentials. Although the application of logarithmic function protects the training algorithm from converges to infinity or not-a-number region, learning has not been observed when using raw feature vectors.

Three initialisation matrices have been experimented with the training algorithm on normalised training set: covariance matrix of the training vectors M_1, identity matrix of the same shape M_2, and matrices with uniformly distributed entries between 0 and 1 M_3. Since gradient descent is a local optimiser, it is expected that the initialisation matrices will converge to different local minima. M_3 is observed with the best training performances, with mAP value larger than standard Mahalanobis distance formulation.

Table 5: Performance of NCA Training

	@rank1	@rank10	mAP
M_1	0.150	0.580	0.070
M_2	0.31	0.77	0.134
M_3	0.665	0.940	0.233

2. Q2

2.1. Experiment 5

K-means and Agglomerative clustering are performed on both the original and unmodified training data. K-means clustering initialised the 32 (number of identities in the training split) cluster centers randomly. The solution depends on the initialisation and the performance score can vary. Therefore K-means clustering was conducted for multiple times and the mean and standard deviation of the scores are calculated. Agglomerative clustering is determinate with a predefined threshold distance. A majority voting algorithm is used to assign label to the cluster. It assigns the cluster with the label which appears most frequently in the

cluster. Since the clusters have different sizes and usually more than one label appear in each cluster, sometimes, a same label may be assigned to more than one cluster. For K-means clustering which has predefined 32 cluster centres, some class labels may not be able to assigned to any cluster because the images of those labels are scattered and clustered with other identities given the initialisation, that means those labels are removed from the data and queries with those labels will never be returned with their true identity. However, by sacrificing some identities that are dispersive in the feature space, in other words, large within-class variance, we can get a better prediction for other identities that are less dispersive. As for Agglomerative clustering, outliers are less likely to be clustered, so diverse images of a identity do not have to be sacrificed by being grouped with other identities but are treated individually, there will be some clusters with cardinality of 1.

When evaluating the labelling performance, each image takes its turn to be submitted as an query, all of the other images are attached new labels. A ranklist of its closest points is generated and the scores are calculated according the true label of the query image. Figure 4 compares the @rank1 accuracy of the two clusterings applied on the original and normalised feature vectors. It can be seen that even the best

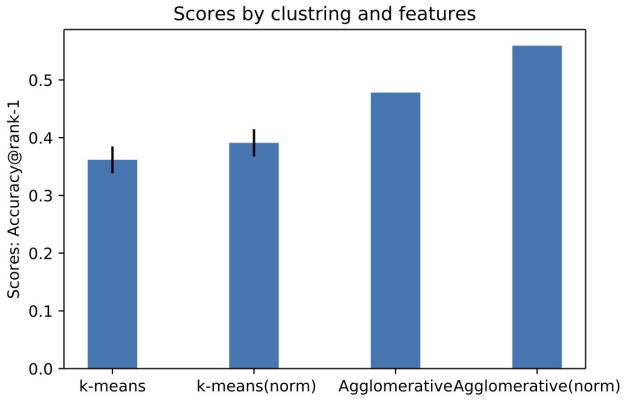

Figure 4: @rank1 accuracy of clustering

case K-means clustring is not as good as the Agglomerative clustering. And applying on normalised vectors is always better than unmodified ones.

2.2. Experiment 6

Using Agglomerative clustering and the distances between images and cluster centres, the images can be represented in the form of Fisher vectors. The number of clusters obtained from the clustering are much smaller than the original dimension of the feature vectors, hence Fisher vectors prove to be an effective of way of reducing the feature dimensions.

The basic representation is to encode the images into arrays of distances between them and each cluster centres. To compare the effect of this representation compared to the original image, l_2 retrieval and cosine similarity retrievals are performed on the transformed test images. The result, though, shows that the retrievals show much lower accuracy at 0.103 and 0.126 respectively. However, on transforming training images to the Fisher vector representation, Mahalanobis distance standard retrieval shows a relatively high mAP at 0.245, with the same distance on Fisher vectors of normalised input at 0.26.

It is possible to apply a softmax on the inverse of the measured distance to scale the distances between 0 and 1. However, due to the numerical stability issue with exponential functions, softmax applied to Fisher vectors can sometime lead to appearance of not-a-number of infinity. Because of the same reason it has not been possible to transform the training images into the same space for Mahalanobis distance measurement. The l_2 retrieval and cosine retrievals in this case are much lower than using raw images, both have mAP 0.137, accuracy around 0.35 at rank 1 and 0.78 at rank 10.

Gaussian Mixture Model is a complex model attempting to summarise the image distribution using a combination of Gaussian centres and their respective variances. With the cluster centres and clustered groups obtained from the agglomerative clustering process, assuming the images can give sufficient description of the clusters they are in, the mean and variance vectors of each Gaussian distribution can be calculated.

It has been observed that the clustering algorithms, because of some pre-defined conditions of termination, may produce clusters with only 1 data point. This poses problem as an attempt to compute the covariance of a single vector results in 0, and renders the computation of the Fisher encoding invalid. To solve the problem, we reject all the clusters with only one image as outliers that do not contribute to the Gaussian Mixture Model. The recorded mAP for l_2 norm and cosine similarity are 0.12 and 0.11 respectively, with accuracy ranging from 0.3 at rank 1 and 0.75 at rank 10.

In agglomerative clustering, it is possible to indirectly change the number of outliers and number of clusters by adjusting the maximum allowed intra-cluster image distance. Weak link is observed between the number of clusters and the retrieval accuracy in the table. In table 7, the mAP value tends to be bigger at non-extreme value of number of clusters.

References

[1] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Infor-*

Table 6: Performance of histograms

Distance Metrics	@rank1 k={51,27,12,best}	@rank10 k={51,27,12,best}	mAP k=51,27,12,best
Euclidean	0.180, 0.205, 0.190, 0.250(k=23)	0.590, 0.595, 0.625, 0.635(k=7)	0.072, 0.075, 0.071, 0.080(k=7)
Chi-square	0.175, 0.180, 0.225, 0.235(k=9)	0.600, 0.625, 0.625, 0.660(k=14)	0.009, 0.009 0.009, 0.024(k=473)
Intersection	0.210, 0.215, 0.200, 0.265(k=78)	0.645 0.645, 0.650, 0.675(k=66)	0.080, 0.079, 0.076, 0.081(k=20)

Table 7: Number of clusters and Accuracy

Number of Clusters	40	44	50	83	90
@rank1	0.67	0.715	0.655	0.71	0.67
@rank10	0.94	0.945	0.945	0.955	0.965
mAP	0.243	0.248	0.2514	0.2456	0.233

mation Processing Systems 17, pages 513–520. MIT Press, 2005.

[2] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. pages 181–185, 05 2002.

Appendices

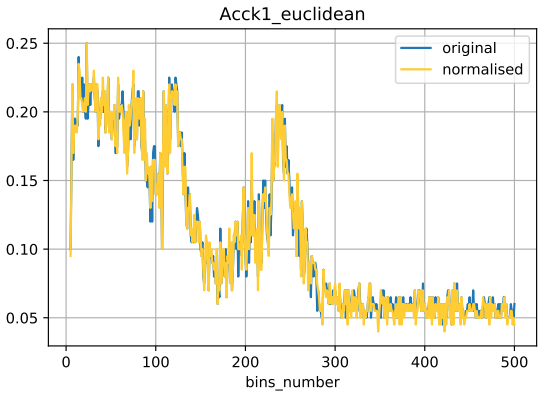

Figure 5: Average Acc_1 of Euclidean Distance

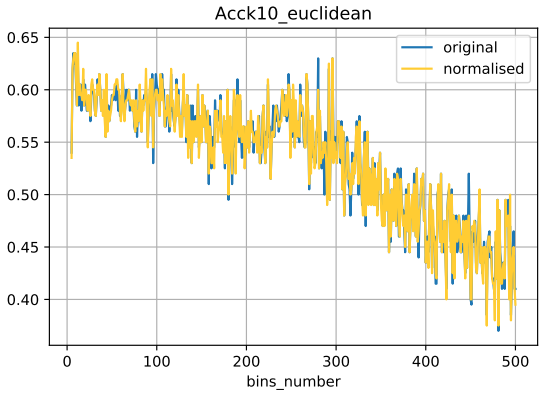

Figure 6: Average Acc_{10} of Euclidean Distance

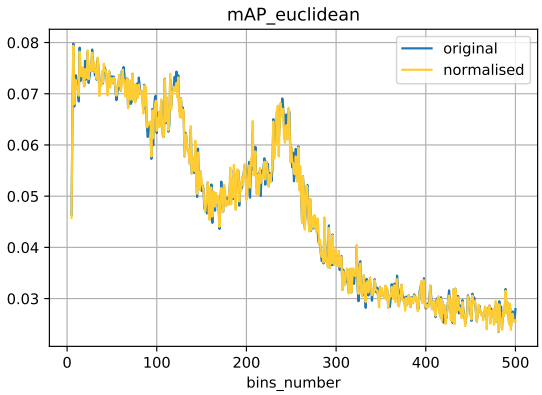

Figure 7: mAP of Euclidean Distance

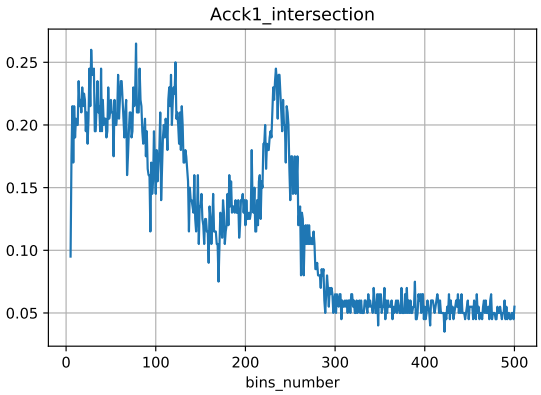

Figure 8: Average Acc_1 of Intersection

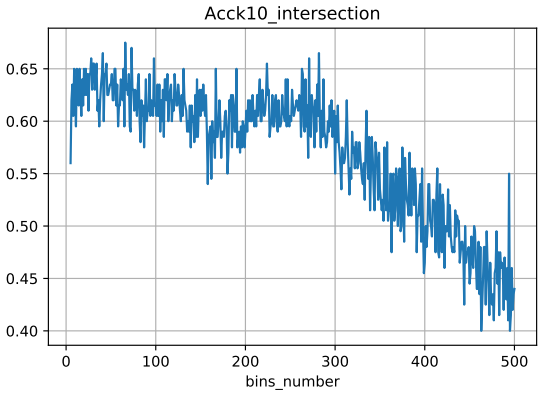

Figure 9: Average Acc_{10} of Intersection

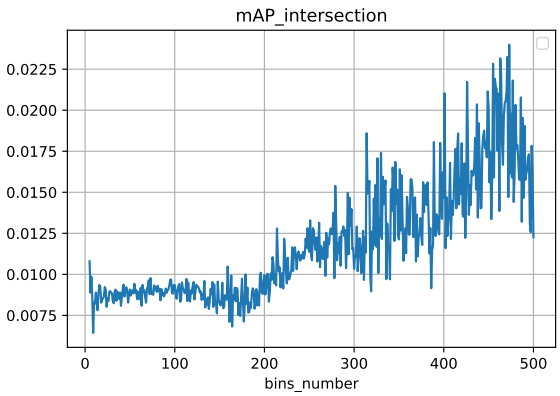

Figure 10: mAP of Intersection

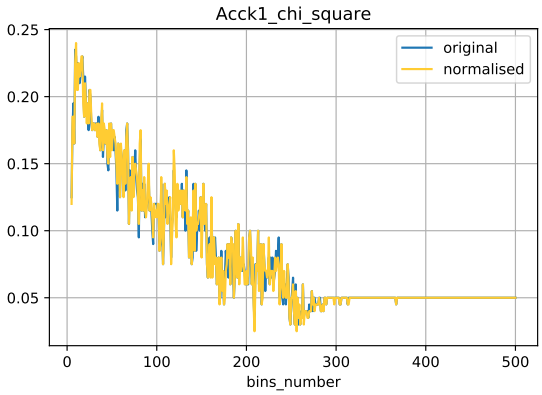

Figure 11: Average Acc_1 of Chi-square Distance

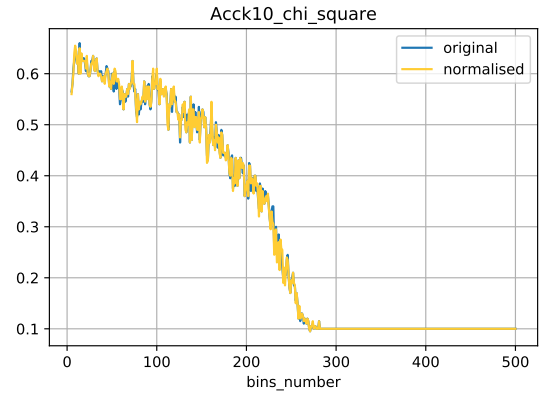

Figure 12: Average Acc_{10} of Chi-square Distance

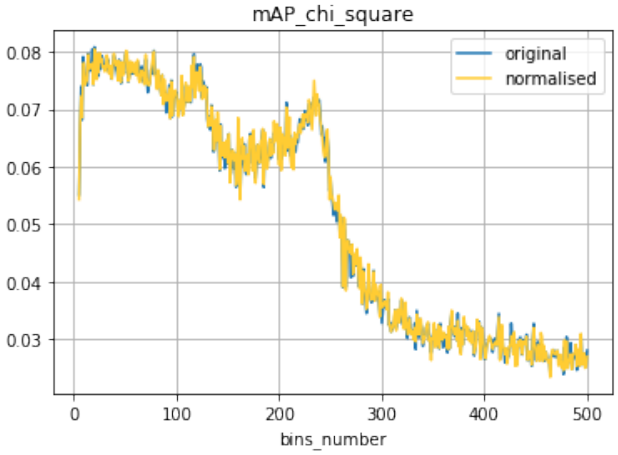

Figure 13: mAP of Chi-square Distance