Pattern Recognition Coursework 2

Iani Gayo
01201287
ig116@ic.ac.uk

Gauri Gupta
01258518
gg2316@ic.ac.uk

1. Question 1 - Distance Metrics

1.1. A: Data

Two sets of data were prepared from the 2576 by 520 face dataset: un-normalized and normalized. The normalised dataset is obtained by dividing the intensties of the images with their respective norms. Normalisation brings all the images to the same scale, thereby reducing the effect of any other variables that could alter the intensities in an image, for instance changes in illumination. Each set of data was partitioned into the training split (320 datapoints) and test split (200 datapoints).

1.2. B: Baseline - Un-learned Standard Metrics

The Baseline experiment was conducted with the following un-learned distance metrics: Euclidean, Cosine, Manhattan, Chebyshev and Minkowski. Table 1 and Table 2 show the three performance scores, mean accuracies at ranks 1 and 10, as well as the mean average precision (mAP) computed from the interpolated precision matrix using all (10) recall levels ([0, 1/9, 2/9 .... 8/9, 1]), for both the normalized and un-normalized test datasets respectively. It can be seen that some metrics, particularly Euclidean, Manhattan and Minkowski respond better to normalized data, while Cosine remains invariant and Chebyshev becomes worse. This clearly shows that no clear dominant similarity emerges for all types of representations of data. Generally, all five metrics obtained similar performance scores with Chebyshev performing the worst.

Metric	Euclidean	Cosine	Manhattan	Minkowski	Chebyshev
Rank 1	63.5 | 68.0	68.0 | 68.0	68.5 | 75.0	63.5 | 68.0	16.0 | 15.0
Rank 10	94.5 | 94.5	94.5 | 94.5	93.5 | 95.5	94.5 | 94.5	62.5 | 56.5
mAP	25.1 | 25.1	25.1 | 25.1	26.6 | 28.1	25.1 | 25.1	10.2 | 9.6

Table 1. Performance scores for un-normalized (left) and normalized data (right) as percentages for Baseline test.

1.3. C: Experiment 1 - Histograms

Each datapoint, an array of pixel intensities representing an image flattened into a vector, was converted to a histogram of pixel intensities with a specified bin size for both the normalized data and un-normalized datasets. The similarity measures used to compare the new histogram features were Earth Mover's Distance (EMD), Jensen-Shannon Divergence (JS), Chi-squared, and Intersection. An EMD value computed for two histograms indicates the cost of transforming one histogram into another. The higher this value, the more costly it is, and the more dissimilar two images are. JS is built by using the Kullback-Leibler Divergence similarity which converts the histograms into probability distributions and measures the difference between them, the higher the JS value the more dissimilar two images are. Chi-squared can be seen as an extension of JS and also tests if two distributions are different. The higher the Chi-squared value for the comparison between two histograms, the more dissimilar the two images are. Intersection measures the overlap between two histograms, the bigger the overlap the more similar the two histograms. The number of bins used in the histograms were changed to see how this could affect performance scores. The number of appropriate bins can be determined using the following equation:

$$k = \frac{max(x) - min(x)}{h} \tag{1}$$

Where h is the bin width that can be chosen. A greater bin width (size) means that more intensities are clustered together in a single feature and the data is more compactly represented, however the resolution of the feature vectors decreases. A smaller bin width results in more bins and more available quantization levels.

From Figure 1, it can be seen that the performance scores vary across different bin numbers. This shows that these metrics are very sensitive to the quantization levels. By considering just the mAP values for both normalised and un-normalised data, it can be seen that the greatest value is observed at bin size = 50. This roughly corresponds well with the general formula used to determine number of bins: $k = \sqrt{n}$ where n corresponds to the number of the observations. In our case, n is 2576 and so the suggested value to use for k is 51 which is consistent with our observations.

Considering just mAP and Rank 1 values, it can be seen that a higher number of bins (b = 100) display small performance scores. This is because with greater bin numbers,

there is greater distinctiveness between points, capturing greater variation between images. For instance, changes in illumination in one image could lead to a large count for a certain intensity value, that's not present in other similar images. Greater bin numbers are less robust to noise present in images, bringing down the performance of the metrics. Hence, we observed a trade-off between robustness and distinctiveness for number of bins used.

For the un-normalised data rank 10 performance scores, there exists a W shape, where the accuracies are higher for the extreme bin numbers (smallest and highest), a peak existing between bin numbers of 40-60 and then smaller intermediate values in between. This suggests that there does exist an optimal bin number that can be used in each distance metric for retrieval of the data that leads to high accuracies with greater number of neighbours considered.

In general, EMD performed the worst in terms of rank 1 and mAP scores but had the highest rank 10 results, whereas chi squared displayed consistently high mAP and rank 1 and 10 accuracies. Intersection scores showed huge variation depending on different bin numbers used, which suggests that it may not be the best metric to use for comparison.

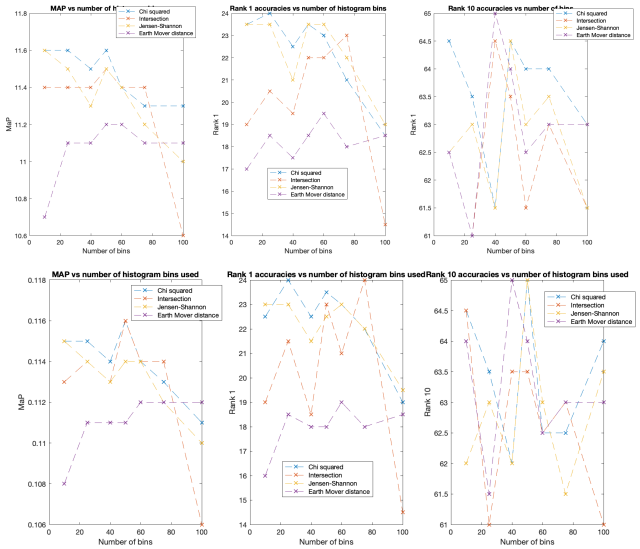

Figure 1. mAP, Rank 1 and 10 accuracies as a function of bin numbers for different metrics used for normalised (above) and un normalised data (below).

1.4. D: Experiment 2 - Generalised Mahalanobis

The Mahalanobis distance is computed using a learned parameter most commonly represented by a matrix A. In this experiment, this parameter was simply the inverse covariance matrix of the training data. This matrix was decomposed via eigenvector decomposition, its linear transformation G was applied to datapoints in the test split, following a Euclidean distance between the projected distance points to compute the similarity between the two images. The aim of this linear transformation is to capture information from the training data and apply it to the test set such that similar points come closer and dissimilar points become further apart. From Table 2, we see that by increasing the number of eigenvectors considered from the covariance matrix, the accuracy scores increase, as more of the covariance matrix is being considered in the Mahalanobis distance calculation. The maximum performance scores are reached by using the full set of eigenvectors. Comparing the Mahalanobis distance metric (full set) to the unlearned distance metrics from the Baseline tests, the performance scores obtained are very similar. However, Cosine and Manhattan metrics outperformed Mahalanobis, which suggests that using the learned inverse covariance from the training data did not provide any additional information that could help with the retrieval of the testing set.

N	16	32	64	128	256	Full set
Rank 1	15.0 | 16.0	31.0 | 23.5	37.0 | 32.0	48.0 | 41.5	59.0 | 56.5	63.5 | 59.0
Rank 10	67.0 | 67.5	72.0 | 72.5	77.0 | 74.0	81.5 | 81.0	87.0 | 84.0	98.5 | 87.5
mAP	10.8 | 11.1	13.8 | 12.8	16.1 | 15.2	19.3 | 18.2	21.4 | 21.7	23.9 | 24.0

Table 2. mAP, Rank 1 and Rank 10 scores for Mahalanobis, for different dimensions used. Un-normalised (left) and Normalized (right).

1.5. E: Experiment 3 - Dimensionality Reduction

The dimensionality of the un-normalized and normalized test face dataset as well as test histogram data was reduced by projecting their datapoints using the LDA coefficients determined by applying PCA and then LDA on the corresponding training split. The Mpca and Mlda values chosen to do this are 50 and 30 respectively as determined by the previous coursework which uses the same dataset. The bin number used for the histogram features is 50 as we observed a peak in mAP for this value (refer to Experiment 2). Once the new test features were produced for a given dataset, the Baseline experiment was ran on it to determine how good the new reduced feature representation is. From Table 3, we see the performance scores for each of the four dimensionality reduced datasets.

The results show that normalised data performed in a similar manner for intensity and histogram PCA-LDA reduced features. However, histogram performance scores are much lower than intensity features. This could be because by using histograms, we're losing distinctiveness between datapoints (lower resolution). By further reducing the dimensionality, there are fewer comparisons to be made during retrieval. Comparing these results with Baseline test and experiment 1, it is observed that dimensionally reduced intensity features still performed quite well and even had higher performance scores. However, the histogram dimensionally reduced data performed much worse. Our conclusion is that there is no need to perform this process on histograms, as histograms are already a compact representa-

Metric	Euclidean	Cosine	Manhattan	Minkowski	Chebyshev
Rank 1	68.0 | 67.5	69.0 | 67.5	65.0 | 64.0	68.5 | 67.5	58.5 | 56.5
Rank 10	97.0 | 96.5	95.0 | 98.0	95.0 |95.5	97.0 | 96	94.0 |95.5
mAP	29.0 | 30.5	30.1 | 32.7	26.6 | 29.5	29.0 | 30.5	24.8 | 24

Table 3. Performance scores for dimensionality reduced unnormalized (left) intensity data and normalised (right) data as percentages for Experiment 3.

Metric	Euclidean	Cosine	Manhattan	Minkowski	Chebyshev
Rank 1	12.5 | 12.5	14.5 | 14.0	10.5 | 11.0	12.5 | 12.5	9.50 |11.5
Rank 10	54.5 | 53.5	56.5 | 57.5	56.5 | 55.5	54.5 | 53.5	50.0 | 51.0
mAP	8.50 | 8.60	8.90 | 9.00	8.20 | 8.20	8.50 | 8.60	8.30 | 8.30

Table 4. Performance scores for dimensionality reduced unnormalized (left) histogram data and normalised (right) data as percentages for Experiment 3

tion of the data, and reducing dimensions further will lose too information about the dataset, making accurate retrieval difficult.

1.6. F: Experiment 4 - Mahalanobis Metric Learning

The matrix A was learned using NCA and LMNN on the training data, as opposed to using the inverse covariance matrix. LMNN works by minimising the distance of same labeled image pairs and shrinks the k nearest neighbours, whilst maximising the separation between all different class pairs. This algorithm makes use of convex nonlinear optimisation approaches. Before using LMNN, we reduced the dimensionality of our data using PCA, to decrease the time taken to learn the metric. NCA is a supervised method which works by associating a probability with a nearest neighbor of a datapoint [1].

The results show that NCA performs better using unnormalised data and receives a maximum accuracy of 94.5% at rank 10. In contrast, LMNN accuracies are better using normalised data, and its accuracy increases even further when using great number of eigenvectors from PCA. Compared with just using the inverse covariance matrix (Experiment 2) of the training set, using LMNN and NCA produced higher performance scores. This is due to the fact that LMNN and NCA are both supervised, and the labels provided allows the linear transformation learnt, to bring similar points closer together whilst keeping dissimilar points as far away as possible. This allows the metric learned to be better suited for the particular data we are dealing with.

Considering rank 1 accuracies for Mahalanobis distance metric using LMNN for normalised data showed that Mahalanobis distance outperformed standard distance metrics like Euclidean and Cosine. This demonstrates the great improvement metric learning from the training data has on retrieval performance compared to un-learned metrics.

Metric	NCA	LMNN (Mpca=100)	LMNN (Mpca=256)
Rank 1	63.5 | 31.0	49.5 | 73.0	63.5 | 72.0
Rank 10	94.5 | 77.0	88.5 | 95.5	94.5 | 95.0
mAP	25.1 | 17.0	24.2 | 35.1	25.1 | 36.2

Table 5. Performance scores for learned Mahalanobis for unnormalized (left) and normalized(right) data as percentages for Experiment 4.

Clustering method	K-means	Agglomerative
Unnormalised test	34	34.5
Normalised test	32	35.5

Table 6. Rank 1 accuracies for different clustering algorithms for un-normalised and normalised test split.

2. Cluster based representations

2.1. Experiment 5: Kmeans vs Agglomerative clustering

Clustering is an unsupervised task which requires partitioning the datapoints into a set number of groups, based on similar features they share.

K means clustering works by randomly initialising a known number of cluster centroids. The datapoints are then assigned according to their nearest centroid. The cluster centres are recomputed by taking the average of all the datapoints assigned to that centre. Clustering is repeated until a certain condition is met, for instance when there's no longer a change of class membership. Agglomerative works with no prior assumption of how many clusters there are to be formed. It works by comparing the distance between all possible pairs of points. Points that are closest to each other are merged and replaced by their average. The process is repeated until a certain threshold is met, i.e. if the distances between points are larger than a set threshold.

To determine which clusters represent which class, a Hungarian mapping algorithm [2] was used. Clustering is performed on both un-normalised and normalised tests and the performances are reported in Table 6. The results show that agglomerative performed better for both sets of testing data. This could be because that the bottom-up approach taken by agglomerative is more robust to outliers, by not randomly initialising centroids. Instead, a better cluster is formed through pairwise comparisons made between datapoints. However, this approach could be computationally expensive and is not recommended for bigger datasets. Kmeans also performs similarly, and could be sufficient given that we know the number of clusters that should be formed.

2.2. Experiment 6: New representations of data

In this experiment, we represent the data in different ways. Agglomerative clustering is performed on the training split and cluster centroids are obtained and used as a codebook. One new representation of the test data is to take

the euclidean distance to each centroid. With 32 centroids initialised, each image is represented as a 32x1 vector as opposed to the original 2576x1. Another representation of the images is to take the softmax probability of the inverse distance to cluster centres. This would give a probability distribution of the proximity of the image to each cluster centroid. Initially centroids are obtained for 32 known clusters, as this is how many clusters we know exist. The results are displayed in Tables 7 and 8.

Metric	Euclidean	Cosine	Manhattan	Minkowski	Chebyshev
Rank 1	33.0 | 35.5	33.5 | 42.5	32.5 | 37.0	33.0 | 36.5	33.5 |38.5
Rank 10	70.0 | 75.5	75.5 | 79	68.0 | 72.0	70.0 | 75.5	74.5 | 81.0
mAP	14.8 | 16.1	15.8 | 17.0	14.6 | 15.9	14.8 | 16.1	15.1 | 16.6

Table 7. Performance scores for features represented as distance to cluster centroids.

Metric	Euclidean	Cosine	Manhattan	Minkowski	Chebyshev
Rank 1	34.5 | 42.5	34.5 | 42.5	34.5 | 42	34.5 | 42.5	32.5 | 41.5
Rank 10	74.5 | 78.0	74.5 | 78.0	76.5 | 75.5	74.5 | 78.0	73.0 | 78.0
mAP	15.9 | 16.3	15.9 | 16.3	15.8 | 15.8	15.9 | 16.3	15.0 | 16.8

Table 8. Performance scores for features represented as softmax probabilities of inverse distance

The results show that generally performance scores for the softmax probability representations are higher, compared with distance to centroid representations. This could be because by considering the probability distributions, we consider the likelihood of a point belonging to each specific cluster as opposed to comparing direct observations of differences akin to association strengths seen in the GMM method. This could account for any uncertainties in the data.

2.3. Fishervectors

Gaussian mixture models (GMM) were computed for the training data and the means were intialized with the clusters centroids resulting from agglomerative clustering. Fishervectors of the corresponding test data were constructed from the means, diagonal covariances, weights and association strengths obtained from the fitted GMM model. Hence, the final dimensions of the fishervector feature representation of test data was 32x2x2576 by 200. This feature representation is larger as opposed to the earlier two representations, however it produces better performance scores due to the fact that the first (similarity to cluster center) and second order statistics (in which direction test datapoint is placed within GMMs) of the GMMs are considered. Furthermore, it was observed that combining the un-normalized data and normalized data produce even better performance scores consistently. Clustering was done using the normalized training data however, the cluster centroids themselves were computed using the data from the un-normalized training data. The performance scores for the Euclidean metric were the following: rank 1 - 57.5, rank 10 - 86.5, and mAP - 20.3. Vice versa was also done and the performance scores of the

Metric	Euclidean	Cosine	Manhattan	Minkowski	Chebyshev
Rank 1	48.5 | 42.5	51.0 | 47.0	50.0 | 48.5	48.5 | 42.5	30.5 | 23.0
Rank 10	70.0 | 67.0	72.5 | 74.0	70.5 | 71.5	70.0 | 67.0	63.5 | 62.5
mAP	16.2 | 13.4	17.2 | 15.9	16.4 | 14.6	16.2 | 13.4	12.9 | 10.9

Table 9. Performance scores for un-normalized and then normalized data as percentages for Baseline test.

Euclidean metric were the following: rank 1 - 60.5, rank 10 - 83.0, and mAP - 19.1. Both combinations produced better performance scores compared to the Euclidean scores of solely the normalized data suggesting that a combination of un-normalized data and normalized data to used to determine cluster centroids and ultimately new test data feature representations is favorable.

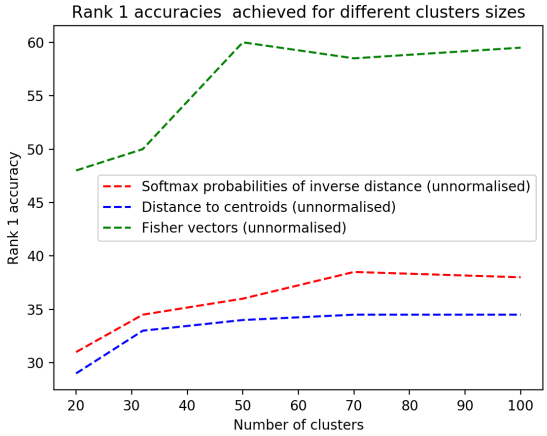

Figure 2. Comparison of rank 1 accuracies for 3 different feature vectors, vs number of clusters used as a codebook

Upon experimenting with the cluster sizes for all new representations it was determined that a smaller number of clusters (20) produces the worst performance scores due to underfitting, and higher number of clusters start producing better performance scores which is to be expected as the features are represented with greater dimensions and more specificity with respect to the training set. This is because clustering is done more meticulously within the training set and has an effect of making retrieval more rigorous, i.e. accounting for the fact that cluster members of one cluster need to have more factors in common which is reflected in the new represented test set.

From Figure 2, it can be seen that using Fishervectors as feature representations for the test data by far outperformed softmax probabilities of inverse distances to centroids and the distance to centroid method obtaining accuracy close to 60% for cluster number 50. This is comparable to the performance scores of using the raw intensity values seen for the un-learned metrics for the Baseline test. Softmax probabilities can be used to initialized the GMM model as well which would give even better performance scores potentially outperforming raw intensities.

3. References

1.https://en.wikipedia.org/wiki/Neighbourhood-components-analysis
2.https://smorbieu.gitlab.io/accuracy-from-classification-to-clustering-evaluation/
3.http://what-when-how.com/computer-visionimaging-and-computer-graphics/fisher-vectors-beyond-bag-of-visual-words-image-representations-computer-visionimaging-and-computer-graphics-part-1/