Pattern Recognition with Distance Metrics

Georgios Gryparis
Imperial College London
00598177
gg1409@ic.ac.uk

Paul Courty
Imperial College London
01247148
pc2816@ic.ac.uk

1. Distance Metrics

The provided dataset is a $D \times N$ matrix, containing $N = 520$ face images, each of size $D = 46 \times 56 = 2576$ pixels. The N images describe 52 people with 10 images per person. The dataset was partitioned in training ($\mathbf{X}_{\mathbf{train}}$) and testing ($\mathbf{X}_{\mathbf{test}}$) datasets, comprised of the first 320 images (classes 1-32) and the last 200 images (classes 33-52) respectively. A second representation was produced, by normalizing each image to unit norm L2 ($\mathbf{Xn}_{\mathbf{train}}$, $\mathbf{Xn}_{\mathbf{test}}$).

1.1. Baseline Retrieval

Retrieval was performed using kNN on $\mathbf{X}_{\mathbf{test}}$ and $\mathbf{Xn}_{\mathbf{test}}$ using standard non-learned metrics for pixelwise image similarity measurement: Manhattan, Euclidian, Chessboard, Cosine Similarity and Normalized Cross-Correlation. These experiments were evaluated using mean accuracy @rank1, @rank10 and mean average precision.

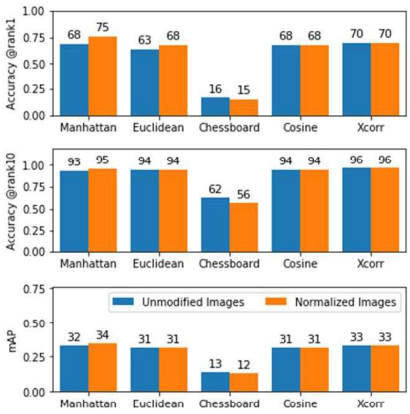

Figure 1. Scores for baseline retrieval

For $\mathbf{X}_{\mathbf{test}}$ and $\mathbf{Xn}_{\mathbf{test}}$, best performance was observed for normalized Cross-correlation and Manhattan distance respectively, for all three performance scores. Note that at each pixel location, there is large variance of intensities. This guarantees a large maximum deviation between images even of the same label, which explain why Chessboard distance performs poorly.

Results for normalized images were generally better. This is due to normalization minimizing differences in average brightness across images. To further improve retrieval performance for pixelwise similarity measurements, histogram equalization is used to adjust contrast within each image (Appendix A).

1.2. Experiment 1

Histograms of pixel intensity for $\mathbf{X}_{\mathbf{test}}$ and $\mathbf{Xn}_{\mathbf{test}}$ were made and used as new feature representations ($\mathbf{HX}_{\mathbf{test}}$, $\mathbf{HXn}_{\mathbf{test}}$). Additionally, histograms of Local Binary Patterns were produced to measure texture ($\mathbf{HX}_{\mathbf{LBP}}$). The number of bins was set to 51 to effectively quantize the data (as it is standard practice to set #bins $= \sqrt{D}$). The range of histograms was set to the minimum and maximum feature value within each dataset (e.g. $0 - 254$ for pixel intensity of $\mathbf{X}_{\mathbf{test}}$), to ensure equivalent quantities were compared. It was experimentally found that this yields better results.

Retrieval was performed, as in the previous experiment, using standard non-learned metrics for histogram similarity measurement: Euclidian, Cosine Similarity, Normalized Cross-Correlation, Earth Mover distance and Intersection.

To assess our choice of #bins=51, the experiment was repeated for all possible #bins (1 to 300). Due to limited computational power the metrics used for assessing optimal #bins were only Euclidian, Cosine and Chi-squared. It was found that while #bins=51 is not always optimal - this will depend on specific metric, specific performance score and dataset (unnormalized, normalized, LBP) - it always produces scores within 10% of the optimal (Appendix B).

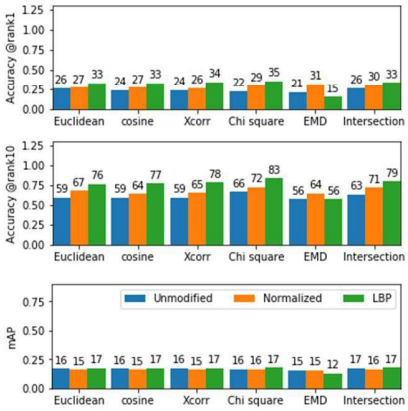

Figure 2. Scores for histogram representations (51 bins)

All metrics investigated perform similarly. As in the previous experiment, $\mathbf{HXn}_{\mathbf{test}}$ outperforms $\mathbf{HX}_{\mathbf{test}}$ due to smaller brightness differences. The LBP representation outperforms $\mathbf{HX}_{\mathbf{test}}$ and $\mathbf{HXn}_{\mathbf{test}}$ for all metrics apart from EMD, which indicates that texture is overall a better discriminator of images than intensity.

Overall, the histogram representations perform poorly compared to baseline results. This is expected as spatial information is lost. To preserve this information, we split each image into 16 sub-images (each with 161 pixels). Sub-image histograms were created (#bins = 13 $\approx \sqrt{161}$) and concatenated to create a feature vector containing both spatial information and intensity frequency.

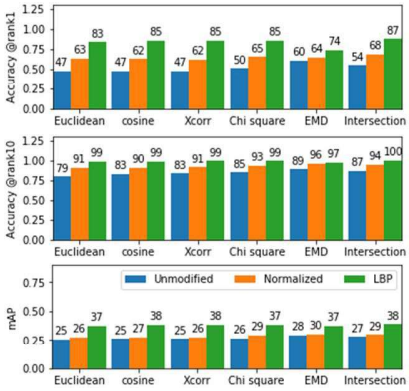

Figure 3. Scores for concatenated histograms

As expected, there is significant improvement compared to both the previous histogram results and the baseline experiment. Splitting LBP achieves the best performance.

1.3. Experiment 2

The Mahalanobis distance metric is used to perform retrieval on both $\mathbf{X_{test}}$ and $\mathbf{Xn_{test}}$. The pseudoinverse ($\mathbf{\Sigma^{+}}$) of the covariance matrix of $\mathbf{X_{train}}$ and $\mathbf{Xn_{train}}$ ($\mathbf{\Sigma}$) was computed and then applied on $\mathbf{X_{test}}$ and $\mathbf{Xn_{test}}$ respectively.

$$d_{MAH}(\boldsymbol{x_1}, \boldsymbol{x_2}) = (\boldsymbol{x_1} - \boldsymbol{x_2})^T \mathbf{\Sigma^{+}} (\boldsymbol{x_1} - \boldsymbol{x_2})$$

To perform dimensionality reduction, we must reduce $\mathbf{\Sigma^{+}}$ to an M by M form ($\mathbf{\Sigma_M^{+}}$), while preserving the variance in our training set (denoted by $\mathbf{X}$). To achieve this, we perform PCA on $\mathbf{X}$ and obtain the normalized ordered eigenvectors $\mathbf{U}$ and the ordered diagonal matrix of eigenvalues $\mathbf{\Lambda}$. Then:

$$\mathbf{\Sigma_M} = \mathbf{U_M \Lambda_M U_M^T} \Rightarrow \mathbf{\Sigma_M^{+}} = \mathbf{U_M \Lambda_M^{-1} U_M^T} = \boldsymbol{G^T G}, \text{ where } \boldsymbol{G} = \mathbf{\Lambda_M^{-1/2} U_M^T}$$

Then, the testing data ($\mathbf{Y}$) is transformed by $\mathbf{Z} = \mathbf{GY}$ (hence, $\mathbf{Z}$ has M rows) and the Mahalanobis distance between $\mathbf{y_1}$ and $\mathbf{y_2}$ reduces to $||\mathbf{z_1}\text{-}\mathbf{z_2}||_2^2$.

Table 1. Scores for Mahalanobis distance metric

	Unmodified Set						Normalized Set					
M	16	32	64	128	256	full	16	32	64	128	256	full
@rank1	0.55	0.63	0.68	0.71	0.69	0.64	0.54	0.59	0.66	0.69	0.63	0.59
@rank10	0.91	0.92	0.97	0.93	0.93	0.90	0.89	0.92	0.94	0.94	0.92	0.88
mAP	0.28	0.30	0.33	0.33	0.32	0.30	0.28	0.30	0.33	0.33	0.31	0.30

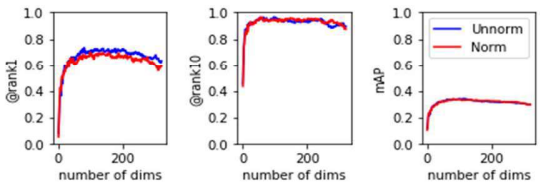

Figure 4. Scores vs number of dimensions (M)

The results are comparable to those in the baseline experiment. At optimal values of M, the scores are slightly better. The optimal values of M differ between performance scores and datasets but are all in the range M=54 to 116 (Appendix C). For M close to the rank of the non-reduced covariance matrix of the training split (320-1=319), overfitting occurs, and the scores drop.

1.4. Experiment 3

In this experiment, PCA-LDA was performed on the training set and the resulting transformation applied to the testing set. This process was conducted on representations investigated in Sections 1.1 and 1.2. Retrieval was performed using the same distance metrics that were used in each respective section. Experiments were conducted for a range of parameters to determine optimal M_{PCA} and M_{LDA}. Due to time constraints, this search was only conducted for Euclidian distance and Cosine similarity (Appendix D).

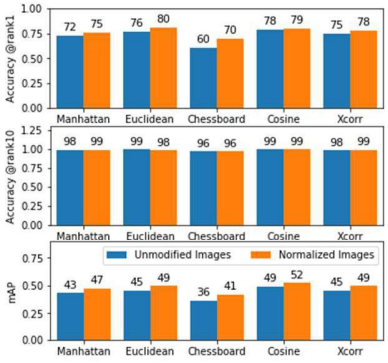

Figure 5. Scores for *transformed* $\mathbf{X_{test}}$ and $\mathbf{Xn_{test}}$ $\binom{M_{PCA_{OPT}}=80}{M_{LDA_{OPT}}=25}$

For the hyperparameters chosen, there is an improvement over the baseline scores. Note that, this is not the case for all combinations of hyperparameters (for example high values of M_{PCA} result in overfitting).

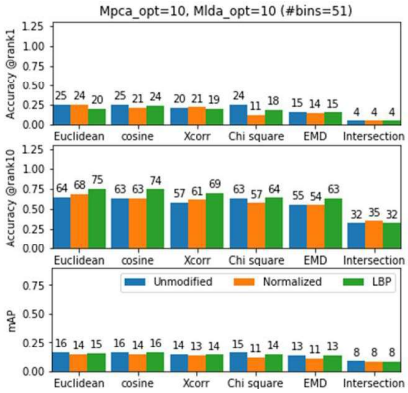

Figure 6. Scores for *transformed* $\mathbf{HX_{test}}$ and $\mathbf{HXn_{test}}$

The results for $\mathbf{HX_{test}}$/$\mathbf{HXn_{test}}$, even for optimal values of M_{PCA}/M_{LDA}, are worse than those in Section 1.2. For EMD, Intersection and χ^2 metrics (which are designed to compare pdfs), this can be attributed to the fact that the feature vectors compared are not comprised of positive elements.

1.5. Experiment 4

Mahalanobis metric learning was used to improve retrieval scores for $\mathbf{X}_{\mathbf{test}}$ and $\mathbf{Xn}_{\mathbf{test}}$. Three techniques of learning the metrics from the training split were explored.

The first method, LMNN, uses the training data to learn a metric that brings each datapoint **x** as close to its k Nearest Neighbors from within its class (called the target neighbors), while also maximizing its distance from "impostors" (datapoints that are of a different class but are within the k Nearest Neighbors of **x**). The algorithm solves a convex optimization problem and, hence, will not converge to local minima. The size of the neighborhood is specified by k. In our experiments we investigate both k=3 (standard practice) and k=9 (the maximum possible size).

The second method, NCA, learns a distance metric that maximizes the leave-one-out classification accuracy in a kNN scheme that is implemented with stochasticity. The stochastic element of this algorithm allows for implementation using gradient descent. However, the optimization problem solved is not convex and, hence, there is likelihood of convergence to local minima. Thus, the results will be greatly affected by initial conditions.

The final algorithm used is LFDA, which combines the ideas behind the supervised Fisher Discriminant Analysis and the unsupervised Locality Preserving Projection to learn a metric that minimizes scatter between the k-Nearest Neighbors within a class and maximize between-class scatter. The value of k used in this experiment is the maximum possible (k=9).

Before application of the algorithms, dimensionality reduction was performed. Scores are plotted below vs number of dimensions. Standard Mahalanobis is added for comparison. NCA is stochastic, thus results were produced by averaging through five iterations. To minimize rounding errors, normalized images are boosted by a factor of 1000.

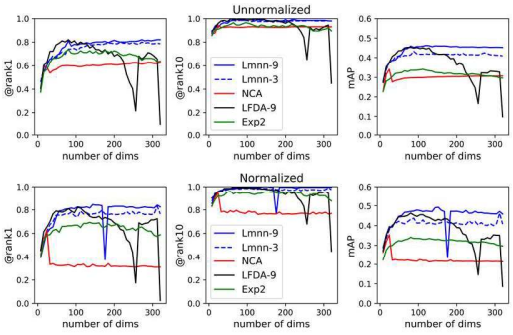

Figure 7. Scores for learnt Mahalanobis Metrics

LMNN with k=9 produces the highest peak but is very computationally intensive. LFDA offers really good results, but the performance drops very steeply as the number of dimensions increases. NCA is the fastest to converge but performs badly. LMNN and LFDA outperform the scores of Experiment 2. The bad performance of NCA can be attributed to convergence to local minima. NCA scores have a steep peak at M=24 and then drop significantly. It is interesting to note that without scaling the normalized dataset by a factor of 1000, the LMNN scores for the normalized data are much worse (rounding errors effect performance). Peak scores in Appendix E.

2. Cluster Based Representations

2.1. Clustering

Two methods for unsupervised data clustering were explored, k-Means and Agglomerative clustering. We first perform clustering on our training data $\mathbf{X}_{\mathbf{train}}$ and $\mathbf{Xn}_{\mathbf{train}}$, and then assign labels to our clusters in a supervised way using the Hungarian algorithm, which is widely used for optimizing label assignment.

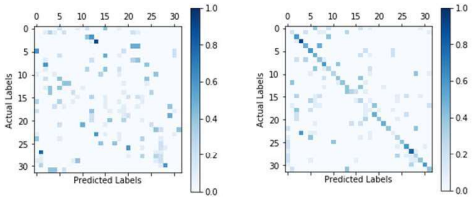

Figure 8. Confusion matrices of $\mathbf{X}_{\mathbf{train}}$ clustered through k-Means (K=32), before (left) and after (right) the application of the Hungarian algorithm

Labeling accuracy @rank1 was measured for both clustering algorithms with different number of clusters (K). Note that as k-Means contains inherent randomness, results were produced by averaging through five iterations. Conversely, for a given distance threshold. agglomerative clustering produces deterministic results.

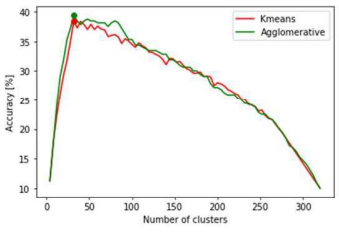

Figure 9. Accuracy @rank1 vs K for $\mathbf{X}_{\mathbf{train}}$

The accuracy for both methods peaks at close to 40%. This can be attributed to the fact that pixelwise intensity is a weak discriminator of images. For more distinctive representations of images, such as SIFT, we would expect much higher accuracy. As expected, the accuracy peaks when K=32 for both $\mathbf{X}_{\mathbf{train}}$ /$\mathbf{Xn}_{\mathbf{train}}$ using either algorithm.

Table 2. Peak Accuracy @rank1 for $\mathbf{X}_{\mathbf{train}}$, $\mathbf{Xn}_{\mathbf{train}}$ (K=32)

	k - Means	Agglomerative
Unnormalized	37.91	39.38
Normalized	40.46	42.19

Results for $\mathbf{Xn}_{\mathbf{train}}$ are better than those for $\mathbf{X}_{\mathbf{train}}$ when using either method. Overall, agglomerative clustering performs better than k-Means, as well as being significantly less computationally intensive.

2.2. Fisher Vectors

2.2.1. Representations Based on Cluster Centers

In this experiment, we use training data cluster centers obtained in the previous section to create new testing data representations. All clustering in this section is performed using the agglomerative algorithm as it scored better in the previous section. The first feature space consists of the Euclidian distances of each image in the testing set to all cluster centers, resulting in K features for each image. The second feature space consists of the softmax probabilities of the inverse of the Euclidian distances of each image to all cluster centers (again K features per image). Retrieval is performed on the new representations of $\mathbf{X}_{\mathbf{test}}$ and $\mathbf{Xn}_{\mathbf{test}}$.

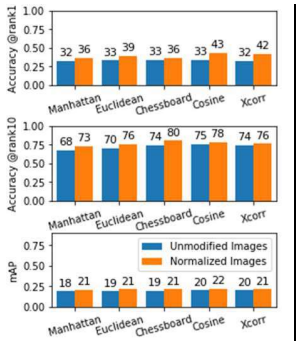

Figure 10. Scores for Rep1 - distance to centers (K=32)

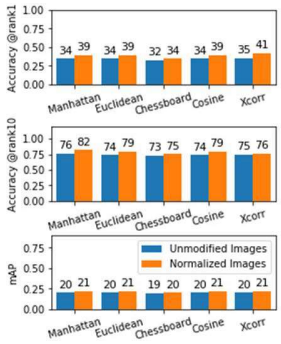

Figure 11. Scores for Rep2 – softmax of inverse distance to centers (K=32)

The results compare poorly to the scores obtained with the baseline representation in section 1.1. The results for the softmax approach are slightly better than those for the first representation. Scores for both new representations of $\mathbf{X}_{\mathbf{train}}$, using Euclidian distance metric for retrieval, are plotted against the number of clusters (K) in Figure 12.

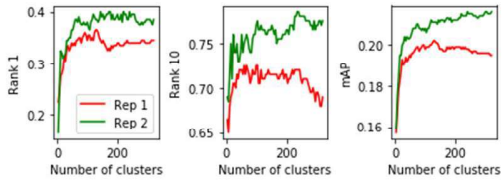

Figure 12. Scores vs K for both representations on $\mathbf{X}_{\mathbf{train}}$

The above figure illustrates that the two new representations perform poorly compared to the baseline for all values of K. Rep-2 (softmax) outperforms Rep-1 for all K. While scores for Rep-1 drop as K tends to the maximum value of 320, scores for Rep-2 increase with K.

The relation between scores and the number of clusters can be understood by considering the following: As K increases over 32, the number of features per image increases but, the cluster-centers become less representative of the training data. Taking the inverse distance and performing the softmax (Rep-2) allows us to exploit the larger number of features without a penalty to accuracy.

2.2.2 Fisher vectors Representation

To obtain a fisher vector representation, we first perform PCA to reduce dimensionality of the training data to M. Then, the data is clustered using the agglomerative algorithm and cluster centroids, variances and relative masses are used to initialize a Gaussian Mixture Model (GMM). The GMM is fitted onto the transformed training data. Due to constraints in computational power, the covariance matrices are set to diagonal.

Once the GMM converges, the fisher vector representation of the testing set is constructed (with the signed squared root operator applied to each vector, which is then subsequently L2 normalized).

It is important to note, that we must initialize the GMM with non-zeros variances. In practice, we found that when $M \geq 8$, there were no zero-variance cases for $8 \leq K < 42$. Retrieval was performed using Euclidian distance metric, and it was found that for $K \in [8,42)$, peak scores occurred when $M = 319$. Due to the inherent randomness in GMMs, we run five iterations of the experiments and average the results. To minimize rounding errors, normalized images are boosted by a factor of 1000.

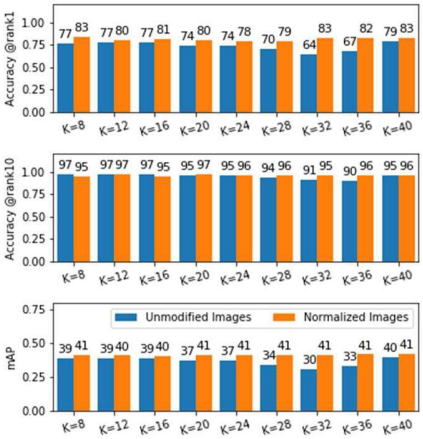

Figure 13. Scores for Fisher vector representations for different K

The Fisher vector representation produces much better results than both representations in the previous section as well as baseline. The results for the normalized images are better than those for the unnormalized images. There is no clear peak at K = 32, which indicates that peak performance of the fisher vector representation does not correspond necessarily with peak clustering accuracy.

Finally, it is important to note that initializing the GMMs with the cluster variances and relative weights, instead of just the cluster centers, yields significantly better results, especially for $K < 26$ or $K > 38$ (Appendix F).

Appendix A

Table 3. Scores for retrieval using equalized $\mathbf{X}_{\mathbf{test}}$

	@rank1	@rank10	mAP
L1	0.81	0.97	0.39
L2	0.76	0.97	0.37
Linf	0.26	0.69	0.15
Cosine	0.76	0.97	0.37
Intersection	0.81	0.97	0.39
Chi Square	0.74	0.98	0.38
EMD	0.16	0.65	0.13
Cross Corr	0.75	0.97	0.37

Appendix B

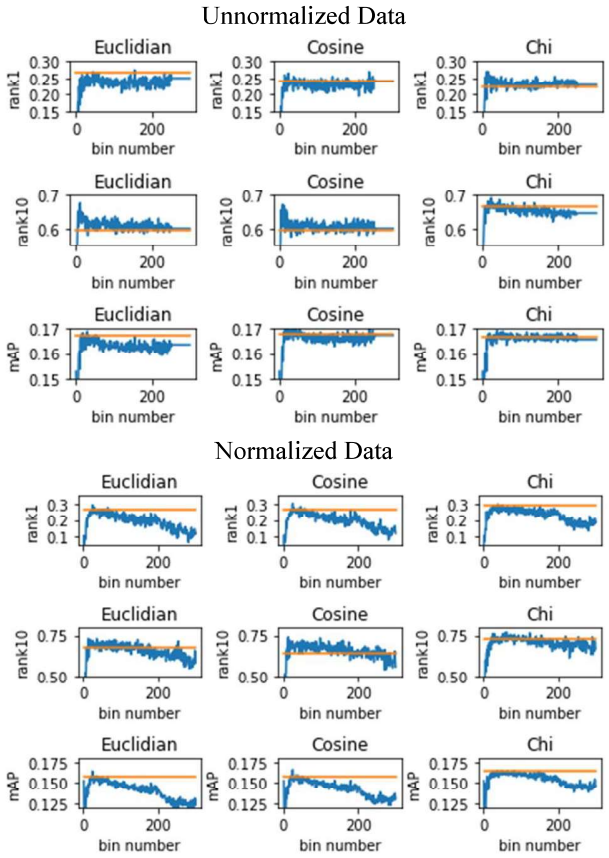

Figure 14. Scores for the histogram representation vs number of bins for Euclidian Cosine and Chi Squared. Orange line represent score for #bins=51

Appendix C

Table 4. Peak Scores for Mahalanobis dimensionality reduction

	Unnormalized	Normalized
Acc @rank1	0.735, for M=80	0.700, for M=99
Acc @rank10	0.965, for M=54	0.965, for M=57
mAP	0.343, for M=116	0.340, for M=80

Appendix D

Euclidian distance on $\mathbf{X}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=64$, $M_{LDA}=25$,
$Acc@rank10_{MAX}$ for $M_{PCA}=120$, $M_{LDA}=13$
mAP_{MAX} for $M_{PCA}=64$, $M_{LDA}=22$

Cosine similarity on $\mathbf{X}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=64$, $M_{LDA}=31$
$Acc@rank10_{MAX}$ for $M_{PCA}=80$, $M_{LDA}=25$
mAP_{MAX} for $M_{PCA}=72$, $M_{LDA}=31$

Euclidian distance on $\mathbf{Xn}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=80$, $M_{LDA}=28$
$Acc@rank10_{MAX}$ for $M_{PCA}=72$, $M_{LDA}=16$
mAP_{MAX} for $M_{PCA}=104$, $M_{LDA}=28$

Cosine Similarity on $\mathbf{Xn}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=80$, $M_{LDA}=28$
$Acc@rank10_{MAX}$ for $M_{PCA}=72$, $M_{LDA}=16$
mAP_{MAX} for $M_{PCA}=104$, $M_{LDA}=28$

All histograms investigated have 51 bins.

Euclidian distance on $\mathbf{HX}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=10$, $M_{LDA}=10$,
$Acc@rank10_{MAX}$ for $M_{PCA}=14$, $M_{LDA}=13$
mAP_{MAX} for $M_{PCA}=8$, $M_{LDA}=7$

Cosine similarity on $\mathbf{HX}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=8$, $M_{LDA}=8$
$Acc@rank10_{MAX}$ for $M_{PCA}=12$, $M_{LDA}=8$
mAP_{MAX} for $M_{PCA}=8$, $M_{LDA}=8$

Euclidian distance on $\mathbf{HXn}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=10$, $M_{LDA}=10$
$Acc@rank10_{MAX}$ for $M_{PCA}=7$, $M_{LDA}=10$
mAP_{MAX} for $M_{PCA}=8$, $M_{LDA}=8$

Cosine Similarity on $\mathbf{HXn}_{\mathbf{test}}$:
$Acc@rank1_{MAX}$ for $M_{PCA}=10$, $M_{LDA}=10$
$Acc@rank10_{MAX}$ for $M_{PCA}=7$, $M_{LDA}=10$
mAP_{MAX} for $M_{PCA}=8$, $M_{LDA}=8$

Appendix E

Table 5. Peak scores for different learnt metrics

	Unnormalized		
	@rank1	@rank10	mAP
LMNN-9	0.82, M=272	0.99, M=72	0.46, M=80
LMNN-3	0.80, M=272	0.99, M=80	0.43, M=272
NCA	0.63, M=256	0.93, M=24	0.34, M=24
LFDA-9	0.83, M=72	0.99, M=72	0.45, M=72
MAHA	0.74, M=80	0.97, M=112	0.34, M=112
	Normalized		
LMNN-9	0.84, M=112	1.00, M=120	0.49, M=112
LMNN-3	0.83, M=312	0.99, M=136	0.47, M=136
NCA	0.63, M=24	0.95, M=24	0.36, M=24
LFDA-9	0.82, M=80	1.00, M=72	0.46, M=80
MAHA	0.7, M=136	0.96, M=56	0.34, M=80

Appendix F

Comparison of fisher vector representation with GMMs initialized only on the cluster centers of the agglomerative clustering algorithm vs representation with GMMS initialized on the centers, covariances and relative weights of the clusters.

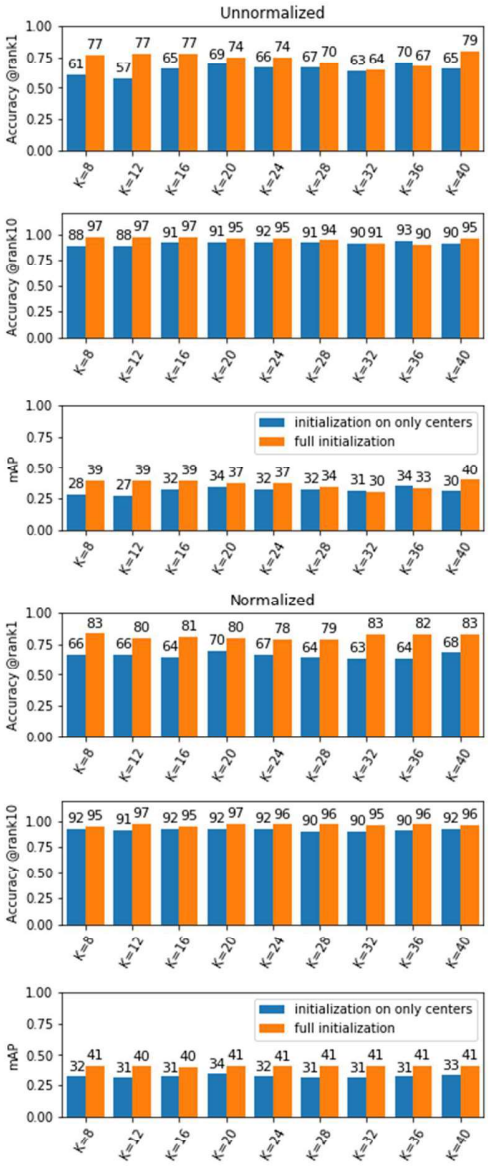

Figure 15. Scores for both Unnormalized and Normalized for two initializations