Pattern Recognition: Coursework 2

Georgios Solidakis: 00975189
Imperial College London
georgios.solidakis15@imperial.ac.uk

Thibaud Lemaire: 01618638
Imperial College London
thibaud.lemaire18@imperial.ac.uk

Abstract

The aim of this report is to investigate the performance enhancement of learning distance metrics on top of a baseline approach using kNN classification. Considering that the kNN algorithm classifies an unlabeled example based on the neighbourhood labels, its performance is dependent upon the distance metric used to calculate the neighbours. Techniques for dimension reduction are implemented in parallel with the Mahalanobis distance metric to enhance the performance. An algorithm which takes examples of similar points as input and learns a distance metric respecting their relationships is finally used. The accuracy of the baseline is only marginally increased but the efficiency and time complexity are greatly enhanced.

1. Problem formulation

The problem we are exploring is on the boundary between classification and clustering. Classification because learning the distance is a supervised problem in which our final goal is to retrieve the identity of a person. Clustering because the testing set does not share the same labels as the training set, thus we need to find a metric (or more generally a feature space) that brings same labelled vectors together while separating them from the others.
The provided data set, *CUHK03* [1], contains 13,164 images of 1,360 pedestrians. The aim is to use the kNN algorithm to perform person re-identification successfully after a query from the data set is issued [2]. Person re-identification is a major challenge in surveillance such as tracking criminals across several cameras. In order to realistically tackle the problem, when queries are issued, the pictures coming from the same camera are never taken into consideration. Under this condition, the retrieval error at different ranks needs to be minimized by the use of different distance metrics and algorithms. The features extracted from the images are split into 3 sets, $T, \mathcal{G}$ and $\mathcal{Q}$ representing the training set, the gallery set and the query set respectively. The feature vector $X \in R^D$ contains the features of dimension D of the N pictures of pedestrians. A ground truth l is provided for each of the pictures $\mathbf{x_i}$ identifying the person shown. Given a specified distance metric $d(x_i, n_k(x_i))$ between a sample $x_i \in Q$ and the k neighbours, the retrieval error of a single query at rank R is defined as: $e = \frac{1}{R} \times neg(l(n_R(x_i)), l(x_i))$, where neg is a function returning the number of times x_i has a different label to one of the k nearest neighbours $n_k(x_i) = min\ d(x_i, x_j),\ x_j \in G$.
In order to minimize this error, the distance metrics and the algorithms implemented must maximize the distance between dissimilar points while minimizing the distance between similar ones. This can be posed as a machine learning optimization problem of the form:

$$max \sum d(x_i, x_j),\ l(x_i) \neq l(x_j)$$

subject to

$$\sum d(x_i, x_j) < \delta,\ l(x_i) = l(x_j)$$

where δ is the maximum threshold allowed between two similar samples, constituting an optimization parameter along with the distance metric d used and parameters of further processing algorithms.

2. Baseline

2.1. kNN

The baseline for evaluating our metrics and techniques is the kNN classification algorithm. The k nearest neighbours of the query image are calculated based on the Euclidean distance and the retrieval error for each query at each rank is calculated as defined above. Figure 6 in the Appendix illustrates examples of queries along with the 10 nearest neighbours calculated.

@rank1	@rank5	@rank10	mAP
0.470	0.669	0.749	0.472

Table 1. Retrieval error at different ranks along with mean average precision(mAP)

Table 1 summarizes the results which are quite satisfying seeing as they are close to the maximum achievable only using the baseline.

2.2. k-Means

In an effort to reduce the retrieval error further, k-means clustering is also applied in our baseline approach aiming to separate the labels better into clusters. In this case, 700 clusters, equal to the number of labels, are computed in the gallery set G. The points are sorted based on their distance to the centroids of the clusters. When an image query is issued, the 5 nearest clusters are computed using kNN along with the cluster centroids, then the points of these clusters are sorted based on the distance to their centroid. The results are summarized on Table 2

@rank1	@rank5	@rank10	mAP	Silhouette
0.383	0.592	0.674	0.445	0.224

Table 2. Retrieval error at different ranks along with mean average precision(mAP) and Silhouette score for k-means baseline.

It is visible that the results are unexpectedly worse. This can be explained by the fact that ideally there should be one cluster per label with 7-8 samples of the same person inside. In reality, the clusters are imperfect with some having only 2 samples and others having 15. This is verified by the silhouette score which is close to 0, indicating overlapping between clusters. Further errors are introduced based on the rank when a query is issued. To calculate rank 10 we need 10 neighbours but the closest cluster might have fewer, in that case the next closest cluster is selected and the points with the smallest distance to the centroid are chosen. The performance can be increased by reinserting pictures of the same person from the same camera in the data set but this is avoided due to overfitting. The real problem being solved is that of re-identifying a person from multiple cameras, not actual classification and hence the data set is kept as it is.

2.3. Distance Metrics

Before more sophisticated algorithms are applied and the set is trained on a metric, a benchmark is needed. Four different distance metrics are applied on the baseline to investigate which one works better for the given data set. The similarity measures used are Minkowski distance(L2), Correlation, Cosine and Mahalanobis. The results are summarized on Table 3 below.

Metric	@rank1	@rank5	@rank10	mAP
Euclidean	0.470	0.669	0.749	0.472
Correlation	0.469	0.664	0.743	0.471
Cosine	0.476	0.670	0.751	0.476
Mahalanobis	0.309	0.455	0.501	0.312

Table 3. Retrieval error at different ranks along with mean average precision(mAP) for different distance metrics.

Overall, there is only a marginal difference using different distance metrics. The Cosine is the only one that outperforms our original baseline. This is owed to the fact that the cosine doesn't depend on the magnitude of the features but is normalized instead. In this way it discriminates against cases where the weighting of a feature is high without conveying significant information. The Mahalanobis distance presented is a "dummy" Mahalanobis distance based on the covariance of the data and has the lowest mAP. The Cross-Correlation did not outperform the baseline either probably due to the high dimensionality. This is motivation to make an effort reducing the features in order to be able to train the metrics on the raw data set.

3. Improved approach

In the next sections, we will consider feature space transformations while using the basic L2 norm in the kNN algorithm. Changing the distance metric used in the kNN is equivalent to finding a feature space in which the samples are spaced in the same way and using the L2 norm [3].

3.1. Dimension reduction

The large size of the features is clear motivation to attempt dimensionality reduction by known pattern recognition techniques. Besides the number of features, dimensionality reduction is needed to prevent overfitting. In this section PCA and LDA are considered along with our kNN classification baseline. Considering the training set and the query and gallery sets share no labels, overfitting can be a huge pitfall for methods like LDA. Before implementing the combination of PCA (for regularisation) and LDA (to maximize the separability), an investigation of the data set for bias is carried out.

3.2. Protocol

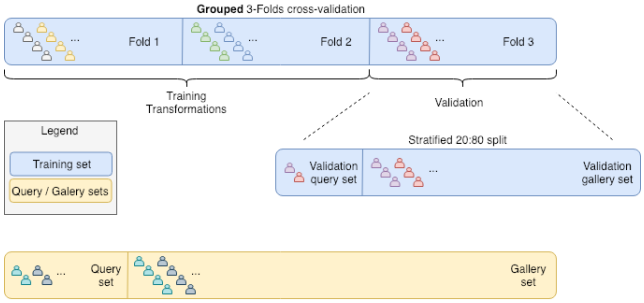

Figure 1. Our approach of cross-validation to make sure images used in validation have not been used for training. Training and Gallery/Query sets are perfectly hermetic.

Structural Risk Minimization is considered in order to optimize the generalization error. The exact conditions of the generalization are reproduced in the validation step of a cross-validation protocol i.e. the training set is split in grouped K-Fold (i.e. one label can only be present in one

fold). During validation, the fold is split again into stratified query and gallery. This protocol is summarized on the figure 1.

3.3. Data bias

To validate the protocol, a grid search on M_{PCA} with cross-validation is performed and the first value with high score is chosen(to prevent choosing a higher M_{PCA} due to the noise) as shown on figure 2. The results are mAP=99.7 for training (fig. 9) and only mAP=49.7 for the gallery (fig. 6). This indicates that the features in the provided data set have been extracted to maximize label separability of the training set meaning they could be biased and potentially overfitted. Furthermore, the Silhouettes of the training and the library sets are calculated as $S_T = 0.516$ and $S_G = 0.200$. This further proves that the training space is biased in the above sense and implies that improving the mAP further on the testing set is hard because it is already capped on the training set.

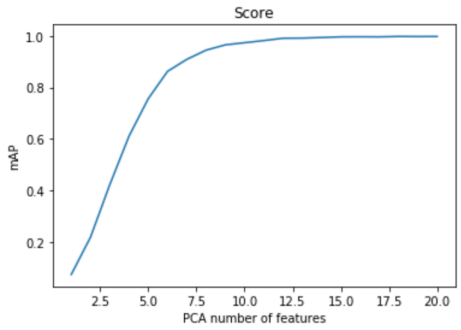

Figure 2. Grid-search with CV of the best M_{PCA} to prevent overfitting. We chose 15 here.

3.4. LDA after PCA

Initially PCA and LDA are considered separately to create a benchmark performance with reduced features. These techniques fit the problem very well due to the large size of the data set and the goal of clustering. PCA is only enough to reduce the dimension but in this case class separation is of utmost importance. LDA full-fills this criterion as it computes directions that will represent the axes that maximize separation between multiple classes. In that way, LDA is already a distance metric learning because it transforms the features into a new space where the data separability is maximized. Considering the results of the two methods are worse than the baselines, an approach using their combination is presented, PCA to regularize the data (prevent overfitting) and LDA to separate the labels. The dimensions of the training set are first reduced to $X_{Train} \in \mathbb{R}^{N_{Train} \times M_{PCA}}$ bounded by the rank of the within-class scatter matrix to ensure a solutions exist. After that LDA is safely applied and a grid search for the best hyperparameters is performed and illustrated in Figure 3 below.

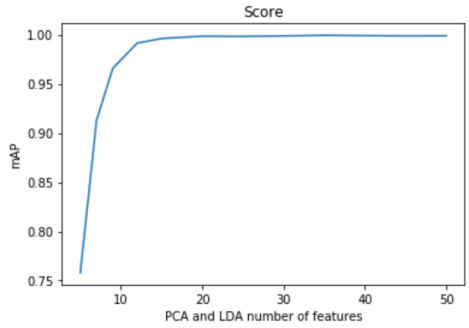

Figure 3. Grid search for optimal parameters M_{PCA} = M_{LDA}

3.4.1 Results

Extracting the optimal parameter $M_{PCA} = 40$ and using it, the mAP score has visibly increased. The final results of this method are summarized in Table 4 below along with those of the benchmarks. Even though there is a visible improvement on the mAP going through the methods mentioned, none of them seems to outperform the baseline. This is owed to the fact that LDA is a supervised algorithm fitting the data based on the labels hence overfitting for the problem of unsupervised learning we are trying to solve. However, it should be noted that since the dimensions of the problem have been vastly reduced, the computational efficiency of the algorithm outperforms that of the baseline hence a significant improvement has been made.

Method	@rank1	@rank5	@rank10	mAP
Baseline	0.470	0.669	0.749	0.472
PCA+kNN	0.419	0.619	0.703	0.425
LDA+kNN	0.427	0.624	0.719	0.432
LDA+PCA+kNN	0.469	0.666	0.751	0.472

Table 4. Results of PCA after LDA along with benchmarks.

4. Improved metric

As mentioned in the problem formulation, we are dealing with a machine learning optimization problem. Furthermore, this problem can be classified as a convex optimization problem that efficient local-minima-free algorithms can be used to solve [3]. A distance metric of the form:

$$d_S(x_i, x_j) = ||x_i - x_j||_S = \sqrt{(x_i - x_j)^T S (x_i - x_j)}$$

is considered. In the general distance metric defined, we try to estimate a matrix S which can be said to parametrize a family of Mahalanobis distances over R^n. Taking S to

be positive definite allows the above to be considered a distance metric by satisfying criteria such as the triangle inequality and non-negativity [4]. The proposed solution to this problem is using the Mahalanobis Metric for Clustering(MMC) [5, 6] along with techniques for dimensionality reduction.

4.1. MMC

The MMC minimizes the sum of squared distances between similar samples while ensuring that dissimilar samples lie further than a certain threshold distance. A supervised version is implemented where pairs of similar samples are created by taking pairs of the same class and dissimilar ones by taking pairs of different classes. This fits the nature of the convex optimization problem and can be solved efficiently. By setting S to be diagonal, different weights are assigned to the different axes giving as one metric of the family. Another metric can be calculated for a full positive definite S. The problem arises from the size of the data set which is very large. The method requires eigendecomposition which is a process of time complexity of order $O(n^3)$, hence training the metric on the whole set would be very inefficient. The two cases of diagonal and full S MMC are combined with PCA and regularized LDA to account for that. PCA and LDA were not better than the baseline on their own but learning the metric might be able to overcome the loss of information introduced by them. The case of full S is not reported due to its inefficiency. Both its results and complexity were much worse than the baseline.

4.2. MMC with Diagonal S

Restricting S to a diagonal matrix allows us to efficiently solve our optimization problem using the Newton-Raphson numerical method. The solution is found in an iterative fashion by calculating the derivative of the function along with a step to continuously move towards the minimum [3].

4.2.1 PCA

In order to overcome the large amount of features, PCA is first applied to the data set reducing its dimensions to 120. After that, the MMC is pipelined with 50000 constraints(sample pairs) and kNN is applied. The results are summarized on Table 5 below while Figure 7 in the Appendix shows query examples and their results.

@rank1	@rank5	@rank10	mAP
0.446	0.644	0.724	0.448

Table 5. Retrieval error at different ranks along with mean average precision(mAP) for MMC after PCA.

The MMC has visibly enhanced PCAs performance but the combination is still outperformed by the baseline. To further enhance these results, regularized LDA is considered instead.

4.2.2 rLDA

The performance of LDA can greatly be increased if it is regularized. The rLDA considered here uses the Ledoit-Wolf lemma to apply automatic shrinkage preventing overfitting as much as possible. A normal LDA is ran first just for benchmarking and its results are presented in table 6 along with rLDAs. Figure 8 illustrates examples of queries and their results using the method described.

Measure	LDA	rLDA
@rank1	0.426	0.471
@rank5	0.624	0.668
@rank10	0.708	0.765
mAP	0.432	0.477

Table 6. Retrieval error at different ranks along with mean average precision(mAP) for MMC after PCA.

The combination of MMC with rLDA has marginally outperformed the baseline. The MMC has greatly enhanced the results of LDA proving that even with the introduced information losses, a good metric is key to good results. The dimensionality reduction along with the ease of the diagonal method computation using numerical methods make this our most efficient and incomplex solution.

5. Conclusion

It was very difficult to overtake the baseline score because of the overfittig trap. Learning a metric able to separate every possible person is a hard task and researchers focus on convolutional neural network approach to achieve it. Among the different approaches implemented, there were only two success cases in terms of accuracy. All the results are compared in figures 4 and 5 in the appendix.

The file submitted for executing training and testing is a Jupyter Notebook so any explanations are included there. It is available at `https://lemaire.io/tvl18_gs2314.zip`.

Appendix

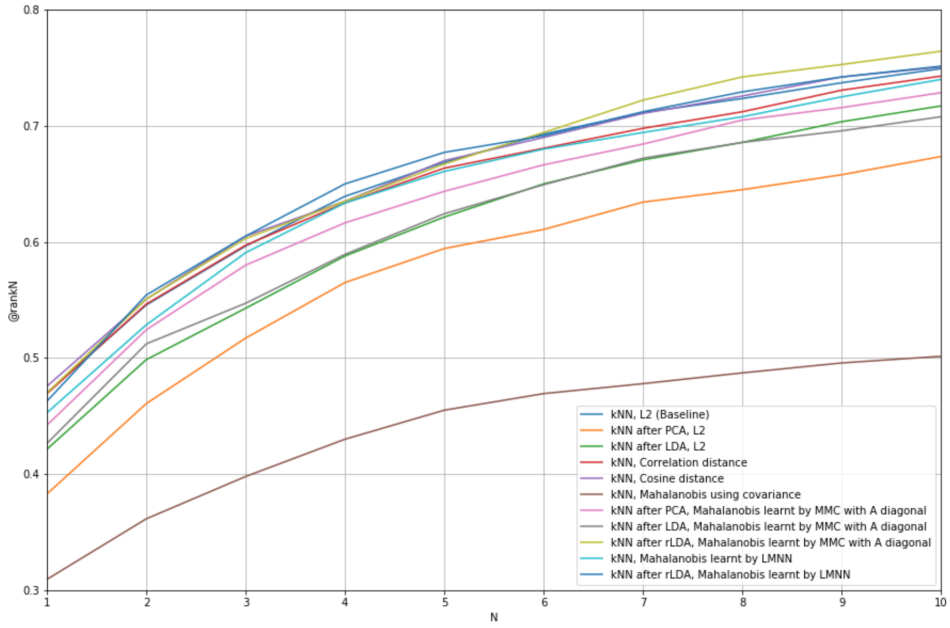

Figure 4. @rankN of the different methods we tried

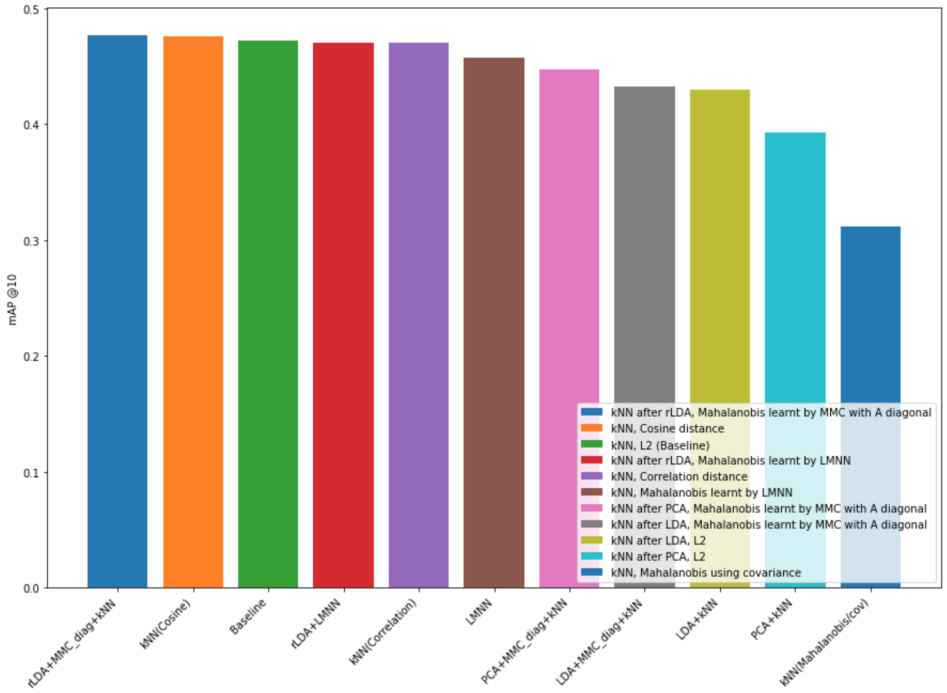

Figure 5. mAP scores of the different methods we tried

Figure 6. Example of gallery using the baseline

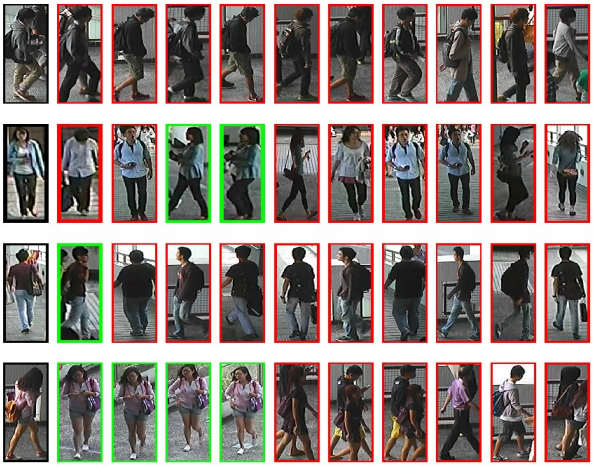

Figure 7. Example of gallery using MMC and PCA

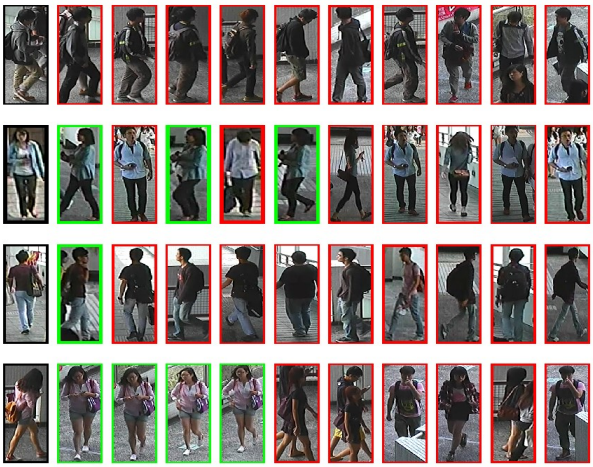

Figure 8. Example of gallery using MMC and LDA

Figure 9. Example of gallery showing the bias on the raw data. Images are taken from the training set. Neither transformation nor learning have been performed.

References

[1] CUHK Data sets
`http://www.ee.cuhk.edu.hk/~xgwang/CUHK_identification.html`

[2] T. Cover and P. Hart, *Nearest neighbor pattern classification*, in IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, January 1967.

[3] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan and Stuart Russell. *Distance Metric Learning, with Application to Clustering with Side-Information*. University of California, Berkeley Berkeley, CA 94720.

[4] Kilian Q. Weinberger, John Blitzer and Lawrence K. Saul. *Distance Metric Learning for Large Margin Nearest Neighbor Classification*. Department of Computer and Information Science, University of Pennsylvania Levine Hall, 3330 Walnut Street, Philadelphia, PA 19104.

[5] MMC documentation from metric-learn package
`https://metric-learn.github.io/metric-learn/`

[6] Metric-learn package on Github
`https://github.com/metric-learn/metric-learn`