Online Mahalanobis Metric Learning for Face Recognition

A Rotation Report
Lampert Group
IST Austria

by Igor Gridchyn

1 Introduction

The *multi class classification problem* is the problem of classifying data samples into more than two classes. The *verification problem* is the problem of deciding whether two data samples belong to the same class or not. In the area of face recognition, data samples are represented by facial images, and different classes are represented by different persons. Face recognition problems are extremely challenging due to high variation in pose, lighting, expression, hair style, occlusions, age etc. However, recent advances in machine learning successfully find application in computer vision and face recognition in particular.
Supervised machine learning algorithms require ground truth information about the training data. There are several kinds of this information, e.g. labelling of the data, i.e. information about class of each data sample, or equivalence constraints, i.e. information about pairs of data samples belonging to the same or to different, however unknown in both cases, classes. The latter can often be easier to obtain.
One more problem arises in case of face extraction from video streams - it is impossible to store all extracted face representation, so application of incremental learning methods makes sense. We describe one such method in this work.

1.1 Basic notions

The notion of distance is utilised in numerous problems of machine learning, including classification, identification, clustering, etc. And the performance of algorithms often highly depends on the quality of corresponding metric [5]. Several *metric learning* algorithms have been developed to learn parameters of metrics which allow parametrisation.

One class of the metrics that allows simple parametric form are Mahalanobis metrics. For a positive semi-definite matrix $M \in \mathbb{R}^{n \times n}$ the Mahalanobis distance between two vectors $x_i, x_j \in \mathbb{R}^n$ and is:

$$d(x_i, x_j) = (x_i - x_j)M(x_i - x_j)^T \tag{1}$$

1.2 Related work

Several methods for learning the matrix M for the Mahalanobis distance have been proposed:

1. In the **Large Margin Nearest Neighbour** (LMNN) [10] algorithm learns such metric, that the distance between samples of different classes is at least by 1 larger than the distance between samples of the same class.

$$\epsilon(M) = \sum_{\{i,j:y_{ij}=1\}} \left[d_M^2(x_i, x_j) + \mu \sum_{\{l,y_{il}=0\}} \xi_{ijl}(M) \right] \tag{2}$$

   where $y_{il} = 0$ if the samples i and l belong to different classes and $y_{il} = 1$ if the samples i and l belong to the same class.

   $\xi_{ijl}(M)$ is a slack variable, which has positive value for impostor x_l and two samples of different class x_i and x_j:

$$\xi_{ijl}(M) = 1 + d_M^2(x_i, x_j) - d_M^2(x_i, x_l) \tag{3}$$

   Thus, distance between samples of the same class is minimised as well as the amount by which impostors from different class invade perimeter of couples of the same class.

2. **Relevant Component Analysis** is described in [1] as a method for learning metric from equivalence constraints, which is knowledge about small groups of data samples (referred to as *chunklets*) known to belong to the same class.
   Verification problem can also be considered as two class classification problem of pairs of data samples. If data samples are represented as feature vectors, then this problem is equivalent to classification of vectors of differences $x_{ij} = x_i - x_j$ between feature vectors.
   For chunklets $C_j = \{x_{ij}\}_{i=1}^n, j = 1 \dots n$ with means m_j authors of [1] use the inverse of chunk let covariance matrix:

$$\hat{C} = \frac{1}{N} \sum_{j=1}^{n} \sum_{i=1}^{n_j} (x_{ij} - m_j)(x_{ij} - m_j)^T \tag{4}$$

   as a Mahalanobis matrix to compute distance between transformed data points: $X_{new} = \hat{C}^{-\frac{1}{2}} X$. Optionally, dimensionality reduction is used.

3. **Linear Discriminant Metric Learning** [4] utilises the probabilistic view on learning a Mahalanobis metric. The probability of a pair of samples belonging to the same class is modelled as

$$p_{ij} = p(y_{ij} = 1 | x_i, x_j; M, b) = \sigma(b - d_M^2(x_i, x_j)) \tag{5}$$

   where $\sigma(z) = (1 + \exp(-z))^{-1}$ is a sigmoid function with a bias term b.

1.3 Online learning

Approaches of online learning are targeted toward developing techniques for updating learned parameters using new training data. Training algorithms usually have large computational cost, so the point is to develop techniques to update learned parameters which will:

- be significantly computationally 'cheaper' than re-running whole learning algorithm.
- not require storing all of the previous data

1.4 KISS Metric Learning

Ko"stinger et. al. [6] have come up with the following Mahalanobis matrix:

$$M = PSD(\Sigma_1^{-1} - \Sigma_0^{-1}), \tag{6}$$

where PSD is operation of projecting the matrix into the space of positive semi-definite matrices and Σ_1^{-1} and Σ_0^{-1} are within and between precision matrices correspondingly:

$$\Sigma_1 = \frac{1}{|S|} \sum_{y_{ij}=1} (x_i - x_j)(x_i - x_j)^T \tag{7}$$

and

$$\Sigma_0 = \frac{1}{|D|} \sum_{y_{ij}=0} (x_i - x_j)(x_i - x_j)^T \tag{8}$$

where S and D are sets of similar and dissimilar pairs correspondingly.

An advantage of this method is that it does not require optimisation techniques compared to other state-of-the-art metric learning methods but rather explicitly represents learned matrix as function of training data.

2 Online Metric Learning

In this work we adapt KISS Metric Learning to the online case. For this we introduce two modifications:

1. computational adaptation for online-learning
2. skipping the PSD projection operation

2.1 Adaptation for online-learning

Let $X^N = \{x_1, \ldots, x_N\}$ be the set of learning data and $Y^N = \{y_1, \ldots, y^N\}$ - the set of corresponding data labels. The basic idea behind online learning is to express learned parameters, in our case (7, 8) for labeled learning data set (X^{N+1}, Y^{N+1}) as an effective function of learning data set (X^N, Y^N), so that learned params can be updated with as little overhead as possible whenever new learning data sample is available. Let C denote the number of different data classes, C_i - subset of $1, \ldots, N$ consisting of data belonging to $i-th$ class, and $c : \mathbb{N} \to \mathbb{N}$ be a labelling mapping, returning the class number of a data sample x_i. For efficient computation this can be extended to:

$$\Sigma_1 = \sum_{i=1}^{N} |C_{c(i)}| x_i x_i^T - \sum_{c=1}^{C} [(\sum_{i \in C_c} x_i)(\sum_{i \in C_c} x_i)^T] \tag{9}$$

$$\Sigma_0 = \sum_{i=1}^{N} [(N - |C_{c(i)}|) x_i x_i^T] - (\sum_{i=1}^{N} x_i)(\sum_{i=1}^{N} x_i)^T + \sum_{c=1}^{C} [(\sum_{i \in C_c} x_i)(\sum_{i \in C_c} x_i)^T] \tag{10}$$

The update rule in case $x_{N+1} \in X_k$ is added to training set for pairwise differences covariance matrices:

$$\Sigma_0^{N+1} = \Sigma_0^N + \sum_{i \notin C_k} x_i x_i^T - (\sum_{i \notin C_k} x_i) x_{N+1}^T - x_{N+1} (\sum_{i \notin C_k} x_i)^T \tag{11}$$

$$\begin{aligned} \Sigma_1^{N+1} = \Sigma_1^N + \sum_{i \in C_k} x_i x_i^T + (|X_k| - 1) x_{N+1} x_{N+1}^T - \\ (\sum_{i \in C_j} x_i) x_{N+1}^T - x_{N+1} (\sum_{i \in C_j} x_i)^T \end{aligned} \tag{12}$$

To update (8) we use Sherman-Morrison formula [7]:

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u} \tag{13}$$

2.2 Positive semi-definiteness of $\hat{M}$

The most computationally expensive operation in the KISS metric learning procedure is projection to the space of positive semi-definite matrices, which requires computing the single value decomposition of the matrix M. It is required because, in general case, the difference of covariance matrix inverses can be a non-PSD matrix. However we observe that, learned for the features described below, the matrix was already positive semi-definite after several iterations on various data-sets. In this case the projection has no effect. Therefore, we adopt a "lazy projection" strategy, where we do not project unless we observe negative distances.

3 Unsupervised Learning

When processing video data, it is possible to extract data for learning for identification problem without having any labelling of the video [4]. Learning data extraction from video is based on two following assumptions:

1. Faces can be tracked between frames to acquire face group of the same person in time.

2. Several faces appearing within the same frame belong to different persons.

Thus, we can extract within-class pairwise differences from time face clusters and between-class pairwise differences from faces appearing in the same frame and, moreover, from time clusters of different persons having common appearance. If $X^1, \ldots, X^n, X^i = \{x_1^i, \ldots, x_{|X^i|}^i\}$, are the face tracks extracted from video and $I = \{(i_1, j_1), \ldots, (i_m, j_m)\}$ is the set of pairs of track indices that intersect in time. Then we can obtain the within-class pairwise differences a $\{x_i^k - x_j^k : k \in \{1, \ldots, n\}, i, j \in \{1, \ldots, |X^k|\}\}$ and between-class pairwise differences as differences $\{x_i^k - x_j^l : (k, l) \in I, i \in \{1, \ldots, |X^k|\}, j \in \{1, \ldots, |X^l|\}\}$.

4 Runtime analysis

To show the effectiveness of online learning, we performed experiments on two public datasets: CAS-PEAL [3] and Buffy [2].

Using the proposed rule to update covariance matrix inverses with a given representation would require $|C_i| + 1$ updates of within-class covariance matrix if an element is added to class i and $N - |C_i| + 2$ updates for between-class covariance matrix. As the Sherman-Morrison update has complexity of $O(d^2)$, where d is dimensionality of feature vector, Mahalanobis matrix update will take $O(Nd^2)$ time, whereas simple inverse of a matrix would take $O(d^{2.807})$ using Strassen algorithm [8].

5 Experimental results

5.1 CAS-PEAL-R1 database subset

Evaluation results of baseline algorithms have been provided for CAS-PEAL-R1 dataset [3]. A first setup uses a simulated stream of faces from CAS-PEAL database. This set is often used as a benchmark for facial recognition algorithms performance evaluation and results for baseline algorithms are available in [3]. The CAS-PEAL-R1 dataset contains 30.900 grayscale 360x480-sized images of 1040 individuals (595 males and 445 females) with varying pose, expression, accessory and lighting (PEAL). For our experiments we select a subset of 6 images per person that reflect the most challenging variation in lighting, expression and occlusions.

Facial feature extraction pipeline From each face image we extract feature vector using the following steps:

1. Face detection (default OpenCV detector)
2. Eye detection (default OpenCV detector)
3. Affine transform by detected eyes, downscaling to size 80x80
4. Lighting normalisation (Single Scale Retinex [11])
5. Gabor filtering - 3 scales, 8 orientations [11, 9]
6. LBP filtering
7. LBP histograms in 16x16 blocks concatenated into final feature vector
8. Dimensionality reduction - final vector of 9600 components reduced by PCA.

Lighting normalisation ensures robustness to variations in lighting and contrast. Invariance to rotations is achieved by geometric normalisation.

Experimental setup We perform learning and evaluation in a simulated online setup. We randomly select images, test the nearest neighbour classifier on them and add them to the training set updating the covariance matrices using equations (11, 12).

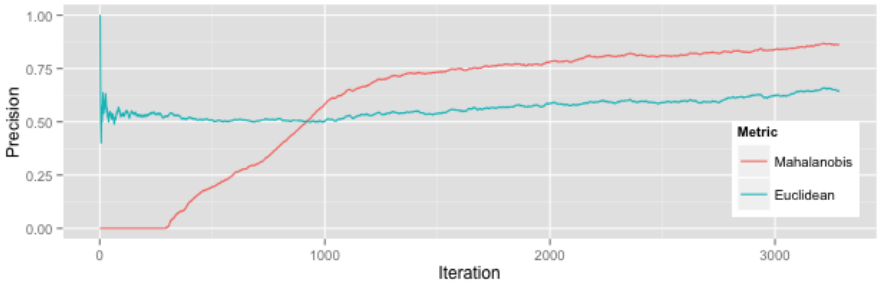

Figure 1: Learning curves, 200 principal components

Learned Mahalanobis metric vs. Euclidean The following figure shows the progress of learning the Mahalanobis metric on 200 principal components. Euclidean metric precision in the end is 64.6%, Mahalanobis metric precision is 86.2%.

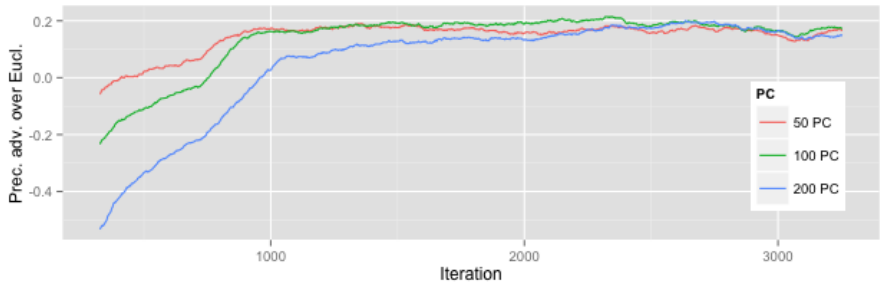

Figure 2: Learning curve for variable number of principal components

Mahalanobis metric for varying number of principal component To understand the dependence on the algorithm performance on the features vector dimensionality, we performed experiments with variable number of principal components. We observed variation in convergence speed and final performance.

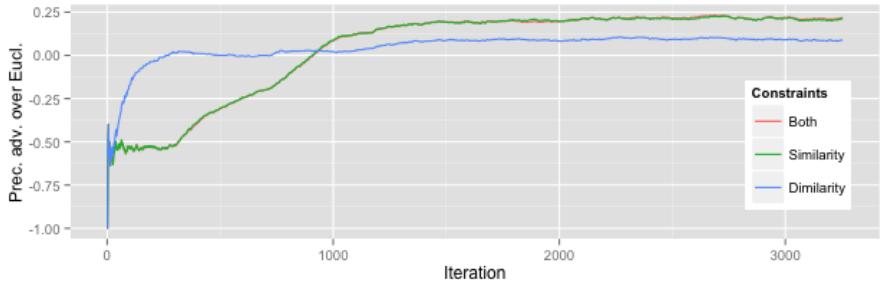

Figure 3: Learning curve for different constraints

Mahalanobis metric learned from within-class only and between-class only covariance matrices (6) has two components. One of them, the within-class covariance matrix inverse, is meant to pull similar example together, while another, the between-class covariance matrix inverse, is meant to push dissimilar examples apart. To understand which effect is dominant, we look at the learning performance of both components individually. As the results show, almost all learning occurs due to information contained in similarity constraints, which supports arguments provided in [1], where Mahalanobis matrix has been learned solely from similarity constraints.

Precision of Euclidean metric for varying number of principal components As we saw, it is computationally costly to run the metric learning with high-dimensional features, so dimensionality reduction by PCA must be

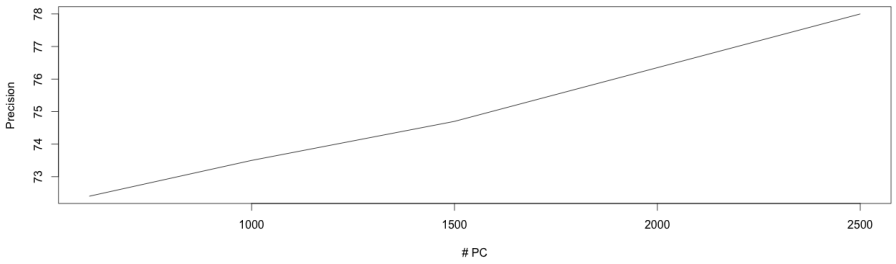

Figure 4: Precision of Euclidean metric for varying number of principal components

applied. An important question is whether the learned Mahalanobis metric will outperform the best Euclidean metric that can be obtained using maximum number of features. The next plot shows performance of the Euclidean metric for different number of principal components.

The performance of Euclidean metric for 2500 components is reached with only 70 principal components if Mahalanobis metric is used. Performance of the Euclidean metric on original 9600 features, which can be considered as reference, reaches 75.6%.

5.2 Automatic character annotation

We used first 6000 out of 24244 annotated faces from 2 episodes of the TV series "Buffy the Vampire Slayer" to explore learning abilities for automatic character annotation. The authors of [2] used the same features.

We used local mean and variance normalised features for automatic character annotation test setup because of better invariance to rotations. The authors of [2] reported that grayscale features had showed better performance than SIFT descriptors because latter can incorporate to much invariance.

In this case Euclidean metric performed quite well, staying above precision of 94%, but learned metric did not perform well if we used both similarity and dissimilarity constraints. Learned from similarity constraints only, the metric performed 2% worse than Euclidean metric.

6 Discussion and future work

In this work we proposed adaptation of KISS Metric Learning algorithm for online learning problem. Though computational complexity of proposed modification is still not enough for online applications, our experiments show that metric outperforming Euclidean distance can be learned from relatively small amount of training data. Precise update of covariance matrices during Mahalanobis matrix update is impossible without having all feature vectors stored,

because we need to know all pairwise differences between new features vector and all previous feature vectors (7, 8). But as not storing all previous data is actually one of the main points of online learning, this limitation should be overcome. Another challenge is to speed up update of precision matrix, for example, by reducing number of Sherman-Morrison updates, which can be also achieved by quantisation.

References

[1] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6, 2005.

[2] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*, 2006.

[3] W. Gao, B. Cao, Sh. Shan, D. Zhou, X. Zhang, D. Zhao, W. Gao, B. Cao, and Sh. Shan Et. Al. The cas-peal large-scale chinese face database and evaluation protocols. Technical report, Joint Research & Development Laboratory, CAS, 2004.

[4] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *International Conference on Computer Vision*, 2009.

[5] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of European Conference on Computer Vision*, 2010.

[6] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2012.

[7] J. Sherman and W. J. Morrison. Abstracts of papers. *The Annals of Mathematical Statistics*, 20(4), 1949.

[8] V. Strassen. Gaussian elimination is not optimal. *Numerical Mathematics*, 13(4), 1969.

[9] V. Štruc and N. Pavešić. The complete gabor-fisher classifier for robust face recognition. *EURASIP Journal on Advances in Signal Processing*, 2010.

[10] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10, 2009.

[11] W. Zhang, Sh. Shan, W. Gao, X. Chen, and H. Zhang. Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In *Proceedings of International Conference on Computer Vision*, 2005.