Metric Learning Approaches for Face Identification

Computer Vision (CSE578) Course Project

Akshay Bankar (2019201011)

Angad Yennam (2019900082)

Face Identification as Metric Learning

- Face identification is a binary classification problem over pairs of face images.
- The confidence scores, or a posteriori class probabilities, for the visual identification problem can be thought of as an object category-specific dissimilarity measure between instances of the category.

Sample Similarity/dissimilarity matrix :

- Squared Euclidean distance :

$$\begin{aligned} d(\mathbf{x}_1, \mathbf{x}_2) &= \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\ &= (\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2) \end{aligned}$$

- Mahalanobis Distance :

$$d_M(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^T \Sigma^{-1} (\mathbf{x}_1 - \mathbf{x}_2) \qquad \Sigma = \sum_{i,j} (\mathbf{x}_i - \mu)(\mathbf{x}_j - \mu)^T$$

Methods for learning :

- Two methods for learning robust distance measures:

a) A logistic discriminant approach which learns the metric from a set of labelled image pairs (LDML) and Its objective is to find a metric such that positive pairs have smaller distances than negative pairs.

b) A nearest neighbor (MkNN) : This method uses a set of labelled images, and is based on marginalising a k-nearest-neighbour (kNN) classifier for both images of a paircomputes the probability for two images to belong to the same class

The author uses following methods to compare his methods :

<u>Large Margin Nearest Neighbors (LMNN)</u> :

- A distance metric learning algorithm for nearest neighbors' classification. It learns a metric that pulls the neighbor candidates (target_neighbors) near, while pushes near data from different classes (impostors) out of the target neighbors' margin.

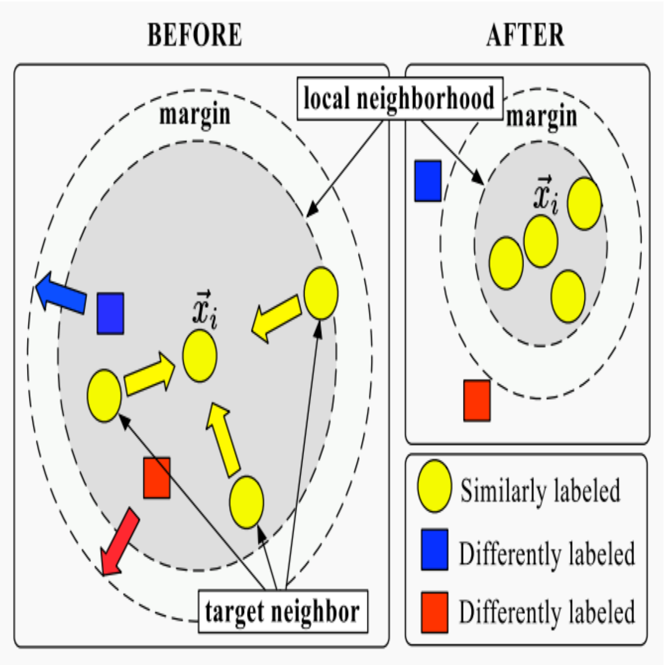
$$\{(\mathbf{x}_i,\mathbf{x}_j) : y_i = y_j, \mathbf{x}_j \text{ belongs to } k\text{-neighborhood of } \mathbf{x}_i\}$$

$$\{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k) : (\mathbf{x}_i,\mathbf{x}_j) \in \mathcal{S}, y_i \neq y_k\}$$

Formulation

$$\min_{M \in \mathbb{S}_+^d, \boldsymbol{\xi} \geq 0} \quad (1-\mu) \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} D_M^2(\mathbf{x}_i,\mathbf{x}_j) \quad + \quad \mu \sum_{i,j,k} \xi_{ijk}$$

$$\text{s.t.} \quad D_M^2(\mathbf{x}_i,\mathbf{x}_k) - D_M^2(\mathbf{x}_i,\mathbf{x}_j) \geq 1 - \xi_{ijk} \quad \forall (\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k) \in \mathcal{R}$$



Information Theoretic Metric Learning :

- An information-theory based distance metric learning algorithm. Given an initial metric, it learns the nearest metric that satisfies some similarity and dissimilarity constraints.
- The closeness between the metrics is measured using the Kullback-Leibler divergence between the corresponding gaussians.

Logistic Discriminant based Metric Learning (LDML) :

- The distance between images in positive pairs to be smaller than the distances corresponding to negative pairs, and obtain a probabilistic estimation of whether the two images depict the same object.
- Using the Mahalanobis distance between two images, model the probability pn that pair n = (i, j) is positive, i.e. the pair label tn is 1, as:

$$p_n = p(y_i = y_j | \mathbf{x}_i, \mathbf{x}_j; \mathbf{M}, b) = \sigma(b - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j))$$

Using maximum log-likelihood to optimize the parameters of the model. The log-likelihood L can be written as:

$$\mathcal{L} = \sum_n t_n \ln p_n + (1 - t_n) \ln(1 - p_n)$$

Identification with Nearest Neighbors (MkNN) :

- Normally, kNN classification is used to assign single data points xi to one of a fixed set of k classes associated with the training data.

- The probability of class c for xi is :

$$p(y_i = c|x_i) = n_{i}^{*}c/k,$$

where ni*c is the number of neighbours of xi of class c.

- Here, we have to predict whether a pair of images (xi, yi) belongs to the same class, regardless of which class that is, and even if the class is not represented in the training data.

- Compute the marginal probability that we assign xi and xj to the same class using a kNN classifier, which equals:

$$p(y_i = y_j|\mathbf{x}_i, \mathbf{x}_j) = \sum_c p(y_i = c|\mathbf{x}_i)p(y_j = c|\mathbf{x}_j)$$
$$= k^{-2} \sum_c n_c^i n_c^j.$$

- The score of Marginalized kNN (MkNN) binary classifier for a pair of images (xi, yi) is based on how many positive neighbour pairs we can form from neighbours of xi and yi.

Implementation :