Comments on “Efficient and Robust Feature Extraction by Maximum Margin Criterion”

Jun Liu, Songcan Chen, Xiaoyang Tan, and Daqiang Zhang

Abstract—The goal of this comment is to first point out two loopholes in the paper by Li et al. (2006): 1) so-designed efficient maximal margin criterion (MMC) algorithm for small sample size (SSS) problem is problematic and 2) the discussion on the equivalence with the null-space-based methods in SSS problem does not hold. Then, we will present a really efficient MMC algorithm for SSS problem.

Index Terms—Efficient algorithm, equivalence, maximal margin criterion (MMC), null space, small sample size (SSS) problem.

I. ORGANIZATION AND PREPARATION

Organization: In this section, we will give some notations and a brief review of maximum margin criterion (MMC) [3] and point out the two loopholes. In Section II, we will propose a really efficient MMC, and then, conclude this comment in Section III.

Let the training set be composed of $c$ classes $C_1, C_2, \ldots, C_c$, the $i$th class have $n_i$ training samples, and $x_j^{i}$ denote the $j$th $D$-dimensional sample from the $i$th class. In total, there will be $n = \sum_{i=1}^{c} n_i$, training samples. In applications such as face recognition, the small sample size (SSS) problem often takes place, namely, $D \gg n$. The within-class scatter matrix $S_w$ and between-class scatter matrix $S_b$ can be denoted as

$$S_w = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n_i} (x_j^{i} - m_i)(x_j^{i} - m_i)^T = H_w H_w^T$$

$$S_b = \frac{1}{n} \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T = H_b H_b^T$$

where $H_w$ and $H_b$ are, respectively, defined as

$$H_w = \frac{1}{\sqrt{n}} [x_1^1 - m, \ldots, x_n^1 - m, \ldots, x_n^c - m]$$

$$H_b = \frac{1}{\sqrt{n}} [\sqrt{n_1}(m_1 - m), \ldots, \sqrt{n_c}(m_c - m)]$$

The $m_i$ is the centroid of the $i$th class and $m$ is the centroid of the training set.

MMC [3] aims at maximizing the average margin between classes, where the interclass margin between the $i$th and $j$th classes can be denoted as

$$d(C_i, C_j) = d(m_i, m_j) - (S(C_i) + S(C_j))$$

where $d(m_i, m_j)$ is defined as the squared Euclidean distance between $m_i$ and $m_j$ and $S(C_i)$ and $S(C_j)$ are, respectively, defined as the traces of the scatter matrix of the $i$th and $j$th classes. After some deduction, the average margin between classes under transformation matrix $W$ is

$$J(W) = \frac{1}{2} (W^T S_b W)$$

Requiring the column vectors in $W$ to be unit vectors, $W$ can be calculated by solving the eigen-equation

$$(S_b - \lambda W)w = 0$$

II. LOOPOLES AND A REALLY EFFICIENT MMC ALGORITHM

A. Loopholes in the Efficient MMC Algorithm

The time and storage complexities for directly solving the eigen-equation (7) are, respectively, $O(D^3)$ and $O(D^2)$, which are very demanding for applications such as face recognition. For example, the face images in the ORL face database [3] have a resolution of $112 \times 92$ (namely, $D = 10304$), thus the time and storage costs are, respectively, in the order of $10^{12}$ and $10^8$, which are very demanding, especially for personal computers.

In [3], Li et al. proposed to obtain a transformation matrix $P \in R^{D \times r}$ that simultaneously diagonalized $S_b$ and $S_w$ as

$$P^T S_b P = \Lambda$$

$$P^T S_w P = I_r$$

and $P$ can be given by

$$P = \Phi \Theta^{-1/2} \Psi$$

where $\Theta \in R^{r \times r}$ is a diagonal matrix whose diagonal entries are the positive eigenvalues of the total scatter matrix $S_b = S_b + S_w$, $r \leq \min\{n-1, D\}$ is the rank of $S_b$, $\Phi \in R^{D \times r}$ contains the $r$ orthonormal eigenvectors of $S_b$ corresponding to these positive eigenvalues, and $\Psi$ is the eigenvector matrix of $\Theta^{-1/2} \Psi^T \Phi \Theta^{-1/2}$.
From (15), one can easily observe that $A \neq B$ and, consequently, neither (14) nor (13) holds. Furthermore, by solving (7) directly, the three nonzero eigenvalues for $S_1 - S_2$ are in fact $11.181, -1.2066,$ and $-0.02269$; rather than $1.0000$, $-1.0000$, and $-1.0000$, as reported in (15). As a result, the calculated $P$ and $2\Lambda - I_r$ are not the right solutions to MMC and the so-designed efficient MMC algorithm in [3] is problematic.

B. Loopholes in Relationship With the Null Space Method

Li et al. argued in [3] that “When the small samples size problem arises, MMC is actually equivalent to LDA + PCA [2] in general.” However, this argument is incorrect. The null-space-based methods such as the LDA + PCA [2] in fact obtain a projection matrix in the null space of $S_w$ as

$$W_{opt} = \arg \max \text{max}_{W^T S_w W = 0} \left| W^T S_1 W \right|$$

(16)

or, equivalently,

$$W_{opt} = \arg \max \text{max}_{W^T S_w W = 0} \left| W^T (S_1 - S_w) W \right|$$

(17)

For discussion convenience, we denote the projection matrix for MMC as $W_{mmc}$ and that for LDA + PCA as $W_{mib}$. Generally speaking, both $W_{mmc}$ and $W_{mib}$ maximize $\text{tr}(W^T (S_1 - S_w) W)$. However, MMC’s projection matrix only has orthogonal constraint, i.e., $W_{mmc}^T S_w W_{mmc}$ is an identity matrix, while LDA + PCA’s projection has additional constraint, namely, $W_{mib}^T S_w W_{mib}$ is an identity matrix and $W_{mib}^T S_w W_{mib} = 0$. As a result, they are distinct theoretically, and $\text{tr}(W_{mmc}^T (S_1 - S_w) W_{mmc}) \leq \text{tr}(W_{mib}^T (S_1 - S_w) W_{mib})$. Since LDA + PCA has additional constraint compared to MMC.

To experimentally show this loophole, we perform experiment on the ORL face database [3], and utilize the first five images of each person to form the training set. The image resolution is $112 \times 92$ and the gray level values of the images are scaled to $[0,1]$. We set the number of projection vectors to 39 (or $c-1$) and employ the subspace distance defined in [1, p. 76] to measure the distance between the subspaces, respectively, spanned by MMC1 and LDA + PCA. The result is that the subspace distance between them is 0.9330 (note that two subspaces are identical only when the subspace distance is 0), which clearly shows that MMC is not identical to LDA + PCA. Furthermore, $\text{tr}(W_{mmc}^T (S_1 - S_w) W_{mmc}) = 47.609$ and $\text{tr}(W_{mib}^T (S_1 - S_w) W_{mib}) = 144.5$, from which we can clearly observe that LDA + PCA cannot achieve the same objective function value as MMC due to the additional constraint $W^T S_w W = 0$.

Finally, in the end of this section, it is worthwhile to note that, although the $P$ computed by (10) is not the correct solution to MMC, it is in fact the solution to the generalized linear discriminant analysis (GLDA) method [4], which was proven equivalent to LDA + PCA [2] when $\text{rank}(S_1) = \text{rank}(S_1) + \text{rank}(S_2)$ in [4]. This partially explains the reasons why MMC was mistaken as equivalent to LDA + PCA in [3] and why the so-called fast algorithm proposed in [3] yielded identical classification performance to LDA + PCA.

C. Really Efficient MMC Algorithm

Despite of the loopholes pointed out in this comment, MMC is a good method that is related to margin between classes. Thus, a really efficient algorithm for MMC in SSS problem is of great importance, and we will offer one in the following.

Due to typically large dimensionality, when implementing MMC, we employ the really efficient MMC algorithm to be given in Section II-C rather than directly solving (7).
Let \( \Phi \) contain the eigenvectors of \( S_i \) corresponding to the zero eigenvalues, and then, the column vectors in \([\Phi \quad \tilde{\Phi}]\) constitute a set of orthonormal bases for the space \( R^D \). According to matrix knowledge [1], any \( w \) in \( R^D \) can be written as

\[
w = \Phi p + \tilde{\Phi} \tilde{p}
\]

(18)

where \( p \) and \( \tilde{p} \) are \( r \times 1 \) and \((D - r) \times 1\) vectors, respectively.

Keeping in mind \( S_i \Phi = 0 \), \( S_w \Phi = 0 \), and \( S_i \tilde{\Phi} = 0 \), and substituting (18) into (7), we get

\[
(S_k - S_w) \Phi p = \lambda (\Phi p + \tilde{\Phi} \tilde{p}).
\]

(19)

Premultiplying \( \Phi^T \) to both sides of (19), we get

\[
(\Phi^T S_k \Phi - \Phi^T S_w \Phi) \Phi p = \lambda \Phi p.
\]

(20)

Premultiplying \( \Phi^T \) to both sides of (19), we get

\[
0 = \lambda \tilde{p}.
\]

(21)

We can set \( \tilde{p} = 0 \), since 1) \( \lambda \) should be as positive as possible in order to maximize MMC’s objective function (6), and thus, it would be better than \( \tilde{p} = 0 \) by using (21); and 2) when the samples are projected by \( \tilde{\Phi} \tilde{p} \), all the training samples concentrate to a common vector and the extracted features by \( \tilde{\Phi} \tilde{p} \) contain no discriminant information, and thus, we can set \( \tilde{p} = 0 \). Now, substituting \( \tilde{p} = 0 \) into (18), we have

\[
w = \Phi p.
\]

(22)

It is easy to observe that the \( ws \) for MMC can be calculated in terms of the following three steps: 1) calculate \( \Phi \), 2) calculate \( ps \) by (20), and 3) obtain \( ws \) from (22). The time complexity of the newly proposed MMC algorithm is analyzed as follows: 1) \( \Phi \) can be obtained in \( O(Dn^2) \), 2) \( \Phi^T S_i \Phi = (\Phi^T H_k)(\Phi^T H_k)^T \) and \( \Phi^T S_w \Phi = (\Phi^T H_w)(\Phi^T H_w)^T \) can be computed in \( O(DnC) \) and \( O(Dn^2) \), respectively, 3) \( ps \) can be obtained in \( O(r^3) \), and 4) \( ws \) can be received from (22) in \( O(DnC) \). As a result, the time complexity is \( O(Dn^3) \), which is much more efficient than \( O(D^3) \) consumed by directly solving (7). Furthermore, since the largest matrix employed in the computation is \( D \times n \), the space complexity is \( O(Dn) \), which is far less than \( O(D^2) \) needed by directly solving (7).

To experimentally verify the correctness of the newly proposed efficient algorithm, we employ it to calculate the projection matrix \( W \) on the two class samples given in Section II-A. Experimental results show that

\[
W = \begin{pmatrix}
-0.4322 & 0.24384 & -0.70655 \\
-0.32817 & 0.50625 & 0.28324 \\
-0.56102 & -0.42688 & -0.27138 \\
0.46483 & 0.51061 & -0.48782 \\
-0.41793 & 0.49122 & 0.33006
\end{pmatrix}
\]

(23)

and the eigenvalues are 11.181, -1.2606, and -0.022609, equal to those directly computed by (7). Thus, the eigenvectors and eigenvalues calculated by the newly proposed efficient MMC are the solutions to (7), and the correctness of the proposed algorithm is experimentally proven.

III. CONCLUSION

In this comment, we point out two loopholes appearing in [3] both theoretically and experimentally, namely, 1) so-designed efficient MMC algorithm for SSS problem is problematic, and 2) the discussion on the equivalence with the null-space-based methods in SSS problem does not hold. Then, we present a really efficient MMC algorithm for the SSS problem.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous reviewers for the constructive comments.

REFERENCES