Reduced Analytical Dependency Modeling for Classifier Fusion

Andy Jinhua Ma and Pong Chi Yuen

Department of Computer Science, Hong Kong Baptist University
Kowloon Tong, Hong Kong
{jhma,pcyuen}@comp.hkbu.edu.hk

Abstract. This paper addresses the independent assumption issue in classifier fusion process. In the last decade, dependency modeling techniques were developed under some specific assumptions which may not be valid in practical applications. In this paper, using analytical functions on posterior probabilities of each feature, we propose a new framework to model dependency without those assumptions. With the analytical dependency model (ADM), we give an equivalent condition to the independent assumption from the properties of marginal distributions, and show that the proposed ADM can model dependency. Since ADM may contain infinite number of undetermined coefficients, we further propose a reduced form of ADM, based on the convergent properties of analytical functions. Finally, under the regularized least square criterion, an optimal Reduced Analytical Dependency Model (RADM) is learned by approximating posterior probabilities such that all training samples are correctly classified. Experimental results show that the proposed RADM outperforms existing classifier fusion methods on Digit, Flower, Face and Human Action databases.

Keywords: Dependency modeling, analytical function, classifier fusion, pattern classification.

1 Introduction

Many computer vision and pattern recognition applications face the challenges of complex scenes with clustered backgrounds, small inter-class variations and large intra-class variations. To solve this problem, many algorithms have been developed to extract local or global discriminative features such as Local Binary Patterns [1], Laplacianfaces [2], etc. Instead of extracting a high discriminative feature, classifier fusion has been proposed and the results are encouraging [3] [4] [5] [6] [7] [8] [9] [10]. While many classifier combination techniques [3] have been studied and developed in the last decade, it is a general assumption that classification scores are conditionally independent distributed. With this assumption, the joint probability of all the scores can be expressed as the product of marginal probabilities. The conditionally independent assumption could simplify the problem, but may not be valid in many practical applications.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part III, LNCS 7574, pp. 792–805, 2012.

In [5], instead of taking the advantage of conditionally independent assumption, the classifier fusion methods are proposed by estimating the joint distribution of multiple classifiers and performance is improved. However, when the number of classifiers is large, it needs numerous data to accurately estimate the joint density [11]. On the other hand, Terrades *et al.* [7] proposed to combine classifiers in a non-Bayesian framework by linear combination. Under dependent normal assumption (DN), they formulated the classifier combination problem into a constraint quadratic programming problem. Nevertheless, if normal assumption is not valid, the results will not be optimal. In this context, Ma and Yuen [10] proposed a linear classifier dependency model (LCDM), which can model dependency under the assumption that the posteriors will not deviate very much from the priors.

Apart from the methods mentioned above, optimal weighting method [4], LPBoost [12] and its multi-class variants [8] aim at determining the correct weighting for linear classifier combination to improve the recognition performance. Since linear classifier combination methods are limited to linear separated systems, Toh *et al.* [6] developed a reduced multivariate polynomial model (RM) to describe the nonlinear input-output relationships for classifier fusion. Although these methods are derived without the conditionally independent assumption, they do not take full advantages of the probabilistic properties in the specific task of dependency modeling.

In this paper, we develop a novel framework for dependency modeling, and propose a method, namely Reduced Analytical Dependency Modeling (RADM) for classifier fusion. Inspired by Product rule [3] (with independent assumption) and LCDM [10] (without independent assumption), we propose to model dependency by analytical functions on posterior probabilities of each feature. With the analytical dependency model (ADM), we give an equivalent condition to independent assumption from the properties of marginal distributions, and show that the proposed ADM can model dependency. Since there may be infinite number of undetermined coefficients in the ADM model, we further propose a reduced form of ADM, based on the convergent properties of analytical functions. At last, under the regularized least square criterion, the optimal RADM is learned by approximating posterior probabilities such that all training samples are correctly classified. The contributions of this paper are two-fold.

- We develop a new framework for dependency modeling by analytical functions on posterior probabilities of each feature. It is shown that Product rule [3] and LCDM [10] can be unified by the proposed ADM framework. On the other hand, we give an equivalent condition when independent assumption is held from the properties of marginal distributions, and show that the proposed ADM can model dependency.
- We propose a novel RADM method for classifier fusion. Since the ADM model may contain infinite number of undetermined coefficients, a reduced form of ADM, which can model dependency as well, is proposed from the convergent properties of analytical functions. After that, an optimal RADM is learned by

a new constraint quadratic programming problem, which minimizes the regularized least square error to approximate posterior probabilities such that all training samples are correctly classified.

The rest of this paper is organized as follows. We first review related works on classifier fusion. Section 3 reports the proposed method. Experimental results and conclusion are given in Section 4 and Section 5, respectively.

2 Related Works on Classifier Fusion

Combining classifiers is one of the strategies to improve recognition performance in general recognition problems. According to Bayesian theory [13], under the conditionally independent assumption, the posterior probability is given by

$$\Pr(\omega_l|\boldsymbol{x}_1,\cdots,\boldsymbol{x}_M)=\frac{P_0}{\Pr(\omega_l)^{M-1}}\prod_{m=1}^{M}\Pr(\omega_l|\boldsymbol{x}_m) \tag{1}$$

where ω_l denotes label, M is the number of feature measurements, $\boldsymbol{x}_m$ is the m-th measurement and $P_0=\frac{\prod_{m=1}^{M}\Pr(\boldsymbol{x}_m)}{\Pr(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_M)}$. Product rule was then derived by (1) in [3]. Moreover, with the assumption that posterior probabilities of each classifier will not deviate dramatically from the priors, Sum rule [3] was induced. Based on Product rule and Sum rule, Kittler *et al.* [3] justified that the commonly used classifier combination rules, i.e. Max, Min, Median and Majority Vote, can be derived. Besides these combination rules developed under Bayesian framework [3], Terrades *et al.* [7] tackled the classifier combination problem using a non-Bayesian probabilistic framework. Under the assumptions that classifiers can be combined linearly and the scores follow independent normal distribution, the independent normal (IN) combination rule was derived [7].

Without conditionally independent assumption, the posterior probability of classifiers can be computed by joint distribution estimation. For example, in [5], Parzen window density estimation was used to estimate the joint density of posterior probabilities by a selected set of classifiers. Since it needs numerous data to ensure that estimation of the joint distribution is accurate [11], Terrades *et al.* [7] proposed to combine classifiers by a linear model under normal distribution assumption. When features are not conditionally independent, the covariance matrix in the normal distribution is not diagonal. In this case, the dependent normal (DN) rule [7] was formulated into a constraint quadratic programming problem, which can be solved by nonlinear programming techniques [14]. Removing the normal distribution assumption on scores, Ma and Yuen [10] proposed to add dependency terms to each posterior probabilities, and expand the product formulation as the linear classifier dependency model (LCDM) by neglecting high order terms, i.e.

$$\Pr(\omega_l|\boldsymbol{x}_1,\cdots,\boldsymbol{x}_M)\approx P_0[(1-M)\Pr(\omega_l)+\sum_{m=1}^{M}a_{lm}\Pr(\omega_l|\boldsymbol{x}_m)] \tag{2}$$

where $a_{l1}, \cdots, a_{lM}$ are the dependency weights. Then, the optimal LCDM model was learned by solving a standard linear programming problem, which maximized margins between genuine and imposter posterior probabilities in [10].

Besides the explicit dependency modeling methods [5] [7] [10], the optimal weighting method (OWM) [4], LPBoost approaches [8] [12] and reduced multivariate polynomial model (RM) [6] can be used to combine classifiers with different kinds of features to improve the recognition performance as well. OWM and LPBoost methods aimed at determining the correct weighting for linear combination by minimizing the classification error and 1-norm soft margin error, respectively. In order to describe the nonlinear input-output relationships, the reduced multivariate polynomial (RM) model was introduced in [6]. Since the number of terms will increase exponentially with the order in the multivariate polynomial, Toh *et al.* [6] proposed to approximate the full polynomial by modified lumped multinomial. Then, the optimal RM model was learned by a weight-decay regularization problem in [6].

3 Reduced Analytical Dependency Modeling

In this section, we propose a novel Reduced Analytical Dependency Modeling (RADM) method to model dependency for classifier fusion. In Section 3.1, we first derive the analytical dependency model (ADM) by unifying Product rule (1) and LCDM (2), as well as give detailed explanation on how the proposed ADM models dependency. Since ADM may contain infinite number of undetermined coefficients, a reduced form of ADM is derived in Section 3.2. Finally, a method to learn the optimal RADM is presented in Section 3.3.

3.1 Analytical Dependency Modeling

Consider a combination problem that, there are M distinct feature descriptors $f_1, \cdots, f_M$ for any object $\mathcal{O}$. Denote feature measurements $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M$ as $\boldsymbol{x}_m = f_m(\mathcal{O})$. The objective of dependency modeling is to estimate the posterior probability $\Pr(\omega_l|\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M)$ for better classification performance. Since the modalities of feature measurements can be different, e.g. $\boldsymbol{x}_m$ can be a vector or a set of points, direct dependency modeling in feature level is difficult. In turn, we consider modeling dependency by posterior probabilities of each feature, $\Pr(\omega_l|\boldsymbol{x}_m)$. Let us denote $s_{lm} = \Pr(\omega_l|\boldsymbol{x}_m)$ and $\boldsymbol{s}_l = (s_{l1}, \cdots, s_{lM})^T$. Suppose the prior probabilities are the same, i.e. $\Pr(\omega_l) = \frac{1}{L}$, where L is the number of classes. With these notations, the Product rule (1) under independent assumption and LCDM (2) for dependency modeling can be rewritten as

$$\begin{aligned}
&\text{Product: } \Pr(\omega_l|\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M) = P_0(L^{M-1}\prod_{m=1}^{M} s_{lm}) = P_0 \cdot h_{\text{Product}}(\boldsymbol{s}_l) \\
&\text{LCDM: } \Pr(\omega_l|\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M) \approx P_0(\sum_{m=1}^{M} a_{lm}s_{lm} + \frac{1-M}{L}) = P_0 \cdot h_{\text{LCDM}}(\boldsymbol{s}_l)
\end{aligned} \qquad (3)$$

As indicated in (3), the Product rule and LCDM model can be formulated as two different functions h_{Product} and h_{LCDM} on posterior probabilities $s_{l1}, \cdots, s_{lM}$. This implies, if we choose a function different from h_{Product} with independent assumption, e.g. h_{LCDM}, the dependency can be modeled. On the other hand, LCDM was proposed under the assumption that posterior probabilities of each classifier will not deviate dramatically from the priors as mentioned in [10]. However, without these assumptions, the function h_l for class ω_l on $s_{l1}, \cdots, s_{lM}$ should be different from h_{Product} and h_{LCDM}. Generally speaking, h_l can be any function which models dependency between feature measurements by the posterior probabilities $s_{l1}, \cdots, s_{lM}$, i.e.

$$\Pr(\omega_l|\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M) = P_0 \cdot h_l(s_{l1}, \cdots, s_{lM}) \tag{4}$$

Analytical functions are very popular and have been employed in Product rule and LCDM. We follow this direction and consider h_l as an analytical function. According to the definition of analytical functions [15], h_l can be expressed explicitly by the converged power series as

$$h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l) = \sum_{|\theta|=0}^{\infty} \alpha_{l\theta} \boldsymbol{s}_l^{\theta} \tag{5}$$

where $\theta = (n_1, \cdots, n_M)^T$, $n_1, \cdots, n_M$ are non-negative integers, $|\theta| = n_1 + \cdots + n_M$, $\boldsymbol{s}_l^{\theta} = \prod_{m=1}^{M} s_{lm}^{n_m}$ and $\boldsymbol{\alpha}_l = (\alpha_{l\mathbf{0}}, \cdots, \alpha_{l\theta}, \cdots)^T$ is weighting coefficient vector in which $\mathbf{0} = (0, \cdots, 0)^T$.

With the analytical dependency model (ADM) given by (5), we further investigate how it can model dependency from the probabilistic aspect. The ADM model (5) can be rewritten according to the order of s_{lm} as,

$$h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l) = \sum_{r=0}^{\infty} g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr}) s_{lm}^{r} \tag{6}$$

where $\tilde{\boldsymbol{s}}_{lm} = (s_{l1}, \cdots, s_{l(m-1)}, s_{l(m+1)}, \cdots, s_{lM})^T$ and g_{lmr} is an analytical function of $\tilde{\boldsymbol{s}}_{lm}$ with coefficient vector $\boldsymbol{\alpha}_{lmr}$. On the other hand, the posterior probabilities can be given by the Bayes' rule [13] as follow,

$$\begin{aligned} \Pr(\omega_l|\boldsymbol{x}_m) &= \frac{\Pr(\boldsymbol{x}_m|\omega_l)\Pr(\omega_l)}{\Pr(\boldsymbol{x}_m)} \\ \Pr(\omega_l|\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M) &= \frac{\Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M|\omega_l)\Pr(\omega_l)}{\Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M)} \end{aligned} \tag{7}$$

Since conditional probability $\Pr(\boldsymbol{x}_m|\omega_l)$ in (7) can be viewed as the marginal probability of the joint density $\Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M|\omega_l)$ over random measurements except $\boldsymbol{x}_m$ [13], we get

$$\Pr(\boldsymbol{x}_m|\omega_l) = \int \Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M|\omega_l) d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_{m-1} d\boldsymbol{x}_{m+1} \cdots d\boldsymbol{x}_M \tag{8}$$

With $P_0 = \frac{\prod_{m=1}^{M} \Pr(\boldsymbol{x}_m)}{\Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M)}$ as mentioned in Section 2 and equations (4) (6) (7), the conditional joint density can be written as

$$\Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M | \omega_l) = \frac{\prod_{m=1}^{M} \Pr(\boldsymbol{x}_m)}{\Pr(\omega_l)} \sum_{r=0}^{\infty} g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr}) s_{lm}^r \tag{9}$$

With notations $s_{lm} = \Pr(\omega_l | \boldsymbol{x}_m)$, substituting the probabilities $\Pr(\boldsymbol{x}_m | \omega_l)$ in (7) and $\Pr(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M | \omega_l)$ in (9) into (8), we get

$$s_{lm} = \int \prod_{i \neq m} \Pr(\boldsymbol{x}_i) [\sum_{r=0}^{\infty} g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr}) s_{lm}^r] d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_{m-1} d\boldsymbol{x}_{m+1} \cdots d\boldsymbol{x}_M \tag{10}$$

According to Abel's Lemma [15], the series in (10) is uniformly converged. Thus, it can be integrated term by term, and equation (10) becomes

$$s_{lm} = \sum_{r=0}^{\infty} G_{lmr}(\boldsymbol{\alpha}_{lmr}) s_{lm}^r \tag{11}$$

where $G_{lmr}(\boldsymbol{\alpha}_{lmr}) = \int \prod_{i \neq m} \Pr(\boldsymbol{x}_i) g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr}) d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_{m-1} d\boldsymbol{x}_{m+1} \cdots d\boldsymbol{x}_M$.

Comparing the left and the right hand sides in (11), the following equations can be obtained,

$$G_{lm1}(\boldsymbol{\alpha}_{lm1}) = 1, \tag{12}$$

$$G_{lm0}(\boldsymbol{\alpha}_{lm0}) = 0, G_{lm2}(\boldsymbol{\alpha}_{lm2}) = 0, \cdots, G_{lmr}(\boldsymbol{\alpha}_{lmr}) = 0, \cdots \tag{13}$$

According to the definition of (6), $g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr})$ is an analytical function similar to $h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l)$ in (5) and vector $\tilde{\boldsymbol{s}}_{lm}$ can be considered as the mappings from feature measurements $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{m-1}, \boldsymbol{x}_{m+1}, \cdots, \boldsymbol{x}_M$ to their posterior probabilities. Therefore, the integration of $\prod_{i \neq m} \Pr(\boldsymbol{x}_i) g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr})$ over feature measurements except $\boldsymbol{x}_m$, which is denoted by $G_{lmr}(\boldsymbol{\alpha}_{lmr})$, is a linear function on coefficient vector $\boldsymbol{\alpha}_{lmr}$. Without calculating the integration, we can observe that $\boldsymbol{\alpha}_{lm0} = \boldsymbol{0}, \boldsymbol{\alpha}_{lm2} = \boldsymbol{0}, \cdots, \boldsymbol{\alpha}_{lmr} = \boldsymbol{0}, \cdots$ is a trivial solution to (13). Substituting this trivial solution into (6), and under the assumption that ADM is symmetric on each s_{lm}, we have equations $h_l(\boldsymbol{s_l}; \boldsymbol{\alpha}_l) = g_{lm1}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr}) * s_{lm}$ for $1 \leq m \leq M$. This implies that each term in the power series h_l contains all variables $s_{l1}, \cdots, s_{lM}$ and the order of each s_{lm} cannot be larger than one. In this case, there is only one non-zero term $\prod_{m=1}^{M} s_{lm}$ in the analytical function h_l. In addition, according to (12), the ADM model becomes $h_l(\boldsymbol{s_l}; \boldsymbol{\alpha}_l) = L^{M-1} \prod_{m=1}^{M} s_{lm}$, which is the Product rule under conditionally independent assumption. This means *independent condition is equivalent to the situation that the solution to* (13) *is trivial.* In other words, if the solution to (13) is non-trivial, the dependency can be modeled. For general analytical functions, the weight vectors $\boldsymbol{\alpha}_{lm0}, \boldsymbol{\alpha}_{lm2}, \cdots, \boldsymbol{\alpha}_{lmr}, \cdots$ are not necessary to be zeros. Consequently, ADM can model dependency by setting non-trivial solution to (13).

3.2 Reduced Form of the ADM Model

The ADM model may have infinite number of coefficients in which direct estimating the coefficient vector $\boldsymbol{\alpha}_l$ is infeasible. In turn, we propose to approximate the ADM model based on convergent properties of the series defined by (5) and (6).

Let us consider the equation (5) again. According to the definition of convergence of series [16], for any positive number ε, there exists a positive integer K, such that $|\sum_{|\theta|=K+1}^{\infty} \alpha_{l\theta} \boldsymbol{s}_l^{\theta}| \leq \varepsilon$. If ε tends to zero, the analytical function can be approximated by the following equation,

$$h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l) \approx h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; K) = \sum_{|\theta|=0}^{K} \alpha_{l\theta} \boldsymbol{s}_l^{\theta} \tag{14}$$

Similarly, since the series defined by (6) is converged, h_l can be approximated by the first $R+1$ terms in (6) as follows,

$$h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l) \approx h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; R) = \sum_{r=0}^{R} g_{lmr}(\tilde{\boldsymbol{s}}_{lm}; \boldsymbol{\alpha}_{lmr}) s_{lm}^{r} \tag{15}$$

where R is a positive integer. According to the analysis in Section 3.1, the approximation in (15) can model dependency by setting solutions to the first R equations in (13) as non-trivial. Therefore, the reduced form of ADM can model dependency. Equations (14) and (15) indicate that the highest order of each term and each variable are K and R, respectively. Combining these two equations, the reduced analytical dependency model (RADM) is given by

$$h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; K, R) = \sum_{|\theta|=0}^{K} \alpha_{l\theta} \boldsymbol{s}_l^{\theta}, \text{ s.t. } \theta = (n_1, \cdots, n_M)^T \text{ and } 0 \leq n_m \leq R \tag{16}$$

Since the model order should be larger than or equal to the variable order, $K \geq R$. On the other hand, if $K > MR$, $h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; K, R) = h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; MR, R)$. This means the RADM model degenerates to $h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; MR, R)$ when $K > MR$. Therefore, the relationship between the model order K and variable order R in RADM is restricted as $R \leq K \leq MR$.

Denote the score vector $\boldsymbol{z}_l = (\boldsymbol{s}_l^{\mathbf{0}}, \cdots, \boldsymbol{s}_l^{\theta}, \cdots)^T$, where $\boldsymbol{s}_l^{\mathbf{0}}, \cdots, \boldsymbol{s}_l^{\theta}, \cdots$ are the terms in (16). With these notations, the RADM model (16) can be written as $h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l; K, R) = \boldsymbol{\alpha}_l^T \boldsymbol{z}_l$. The algorithmic procedure to obtain $\boldsymbol{z}_l$ in RADM for class ω_l is given in Algorithm 1.

3.3 Learning the Optimal RADM Model

The optimal coefficient vector $\boldsymbol{\alpha}_l$ in (16) is determined by a learning process from J training samples $\mathcal{O}_1, \cdots, \mathcal{O}_J$ and their corresponding labels $y_1, \cdots, y_J$. Let us consider the posterior probabilities as, $\Pr(\omega_l|\mathcal{O}_j)$ is equal to one, if $\omega_l = y_j$, and zeros, otherwise. This ensures that all samples are correctly classified.

Algorithm 1. Construct the score vector $\boldsymbol{z}_l$ in the RADM model.

```
Require: Posterior probabilities $s_{l1}, \cdots, s_{lM}$ and model parameters $K, R$;
Set $\mathcal{D} = (0, 1, \cdots, R)^T$ and $\boldsymbol{z}_l = (1, s_{l1}, s_{l1}^2, \cdots, s_{l1}^R)^T$;
for $m = 2, 3, \cdots, M$ do
    Set $\tilde{\mathcal{D}} = (\mathcal{D}, \mathbf{0})$, where $\mathbf{0} = (0, \cdots, 0)^T$ with the same column dimension of $\mathcal{D}$;
    for $r = 1, 2, \cdots, R$ do
        Update $\tilde{\mathcal{D}} = (\tilde{\mathcal{D}}; (\mathcal{D}, r\mathbf{1}))$ which is the column concatenation of $\tilde{\mathcal{D}}$ and $(\mathcal{D}, r\mathbf{1})$,
        where $\mathbf{1} = (1, \cdots, 1)^T$ with the same column dimension of $\mathcal{D}$;
        Update $\boldsymbol{z}_l = (\boldsymbol{z}_l; s_{lm}^r \boldsymbol{z}_l)$ which is the column concatenation of $\boldsymbol{z}_l$ and $s_{lm}^r \boldsymbol{z}_l$;
        Delete the rows in $\tilde{\mathcal{D}}$ and corresponding elements in $\boldsymbol{z}_l$ such that the summations of the rows in $\tilde{\mathcal{D}}$ are larger than $K$;
    end for
    Set $\mathcal{D} = \tilde{\mathcal{D}}$;
end for
return $\boldsymbol{z}_l$.
```

On the other hand, with equation (4), the posterior probability is computed by $P_0 * h_l(\boldsymbol{s}_l; \boldsymbol{\alpha}_l)$. Since P_0 is a parameter depending on $\mathcal{O}_j$, denote p_j as the parameter with respective to $\mathcal{O}_j$. With these notations, we get

$$h_l(\boldsymbol{s}_{jl}; \boldsymbol{\alpha}_l) = q_{jl} = \delta_{jl}/p_j, \quad \delta_{jl} = \begin{cases} 1, & \omega_l = y_j \\ 0, & \omega_l \neq y_j \end{cases} \tag{17}$$

where $\boldsymbol{s}_{jl} = (s_{jl1}, \cdots, s_{jlM})^T$, $s_{jlm} = \Pr(\omega_l | \boldsymbol{x}_{jm})$ and $\boldsymbol{x}_{jm} = f_m(\mathcal{O}_j)$. Denote $\boldsymbol{z}_{jl} = (\boldsymbol{s}_{jl}^{\mathbf{0}}, \cdots, \boldsymbol{s}_{jl}^{\theta}, \cdots)^T$, where $\boldsymbol{s}_{jl}^{\mathbf{0}}, \cdots, \boldsymbol{s}_{jl}^{\theta}, \cdots$ are the terms in (16). The objective function for learning the optimal RADM is given by approximating the distribution (17) under regularized least square criterion [17] as follows

$$\min_{\boldsymbol{\alpha}, \boldsymbol{q}} \sum_{l=1}^{L} \sum_{j=1}^{J} (\boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl} - q_{jl})^2 + b \sum_{l=1}^{L} \|\boldsymbol{\alpha}_l\|^2 \tag{18}$$

where $\| \cdot \|$ denotes the L_2-norm and b is a regularization constant.

As mentioned in Section 2, $p_j = \frac{\prod_{m=1}^{M} \Pr(\boldsymbol{x}_{jm})}{\Pr(\boldsymbol{x}_{j1}, \cdots, \boldsymbol{x}_{jM})}$. Since estimations of probability $\Pr(\boldsymbol{x}_{jm})$ for each m and joint density $\Pr(\boldsymbol{x}_{j1}, \cdots, \boldsymbol{x}_{jM})$ are difficult, direct computation of p_j may not be feasible. Consequently, $p_1, \cdots, p_J$ are treated as undetermined variables. Thus, different from traditional least square regularization [17], optimization problem (18) cannot be solved directly. In this context, we rewrite the objective function (18) as follows, so that it can be solved.

According to (17), $q_{jl} = 0$, if $y_j \neq \omega_l$. The optimization problem (18) becomes

$$\min_{\boldsymbol{\alpha}, \boldsymbol{q}} \sum_{l=1}^{L} [\sum_{y_j = \omega_l} (\boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl} - q_{jl})^2 + \sum_{y_j \neq \omega_l} (\boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl})^2] + b \sum_{l=1}^{L} \|\boldsymbol{\alpha}_l\|^2 \tag{19}$$

Let $\boldsymbol{q}_l = (q_{j_1 l}, \cdots, q_{j_{N_l} l})^T$, such that $y_{j_n} = \omega_l$, where N_l is the number of samples for class ω_l. On the other hand, denote $\mathcal{A}_l$ and $\mathcal{B}_l$ as matrices made up by vectors

$\boldsymbol{z}_{jl}$ for $y_j = \omega_l$ and $y_j \neq \omega_l$, respectively. With these notations and adding a regularization term to $\boldsymbol{q}_l$, the optimization problem (19) can be reformulated as

$$\min_{\boldsymbol{\alpha},\boldsymbol{q}} \sum_{l=1}^{L} [(\boldsymbol{\alpha}_l^T \mathcal{A}_l - \boldsymbol{q}_l^T)(\mathcal{A}_l^T \boldsymbol{\alpha}_l - \boldsymbol{q}_l) + \boldsymbol{\alpha}_l^T \mathcal{B}_l \mathcal{B}_l^T \boldsymbol{\alpha}_l + b * \boldsymbol{\alpha}_l^T \boldsymbol{\alpha}_l + c * \boldsymbol{q}_l^T \boldsymbol{q}_l] \quad (20)$$

where c is a regularization constant for $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_L$. Since the undetermined vector tuples $(\boldsymbol{\alpha}_1; \boldsymbol{q}_1), \cdots, (\boldsymbol{\alpha}_L; \boldsymbol{q}_L)$ are independent with each other with respective to index l. The problem (20) can be broken down into L independent optimization sub-problems, and each of them is written as

$$\min_{\tilde{\boldsymbol{\alpha}}_l} \tilde{\boldsymbol{\alpha}}_l^T \mathcal{H}_l \tilde{\boldsymbol{\alpha}}_l, \text{ s.t. } \mathcal{H}_l = \begin{pmatrix} \mathcal{A}_l \mathcal{A}_l^T + \mathcal{B}_l \mathcal{B}_l^T + bI_{\boldsymbol{\alpha}_l} & -\mathcal{A}_l \\ -\mathcal{A}_l^T & (1+c) I_{\boldsymbol{q}_l} \end{pmatrix}, \tilde{\boldsymbol{\alpha}}_l = (\boldsymbol{\alpha}_l; \boldsymbol{q}_l) \quad (21)$$

where $I_{\boldsymbol{\alpha}_l}$ and $I_{\boldsymbol{q}_l}$ are identity matrices with same dimensions as $\boldsymbol{\alpha}_l$ and $\boldsymbol{q}_l$, respectively.

In order to ensure that the probabilities are positive in the optimal RADM, constraints need to be added to the optimization problem (21). Since $q_{jl} = 1/p_j = \frac{\Pr(\boldsymbol{x}_{j1}, \cdots, \boldsymbol{x}_{jM})}{\prod_{m=1}^{M} \Pr(\boldsymbol{x}_{jm})} > 0$ for $y_j = \omega_l$, the first constraint is set as $q_{jl} \geq \eta$ for $y_j = \omega_l$, where η is a positive number such that $\eta = \min_{j, y_j = \omega_l} q_{jl}$. On the other hand, according to the analysis in Section 3.2, the RADM model approximates (but is not exactly equal to) the posterior probability $\Pr(\omega_l | \mathcal{O}_j)$. This means $\Pr(\omega_l | \mathcal{O}_j) = p_j * (\boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl} + \varepsilon_{jl}) \approx p_j * \boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl}$, where ε_{jl} represents the reminder term close to zero. Since $\Pr(\omega_l | \mathcal{O}_j)$ is positive, $\boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl} \geq -\varepsilon_{jl}$. To avoid introducing extra parameters ε_{jl}, we set the necessary condition that the posterior probabilities are positive as the second constraint, i.e. $\boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl} \geq -\varepsilon_0$, where ε_0 is a small constant such that $\varepsilon_0 = \max_{j,l} \varepsilon_{jl}$.

With these two constraints, the optimal $\tilde{\boldsymbol{\alpha}}_l$ in the RADM model is learned by the following optimization problem,

$$\begin{gathered} \min_{\tilde{\boldsymbol{\alpha}}_l} \tilde{\boldsymbol{\alpha}}_l^T \mathcal{H}_l \tilde{\boldsymbol{\alpha}}_l \\ \text{s.t. i) } q_{ln} \geq \eta, \forall 1 \leq n \leq N_l; \text{ ii) } \boldsymbol{\alpha}_l^T \boldsymbol{z}_{jl} \geq -\varepsilon_0, \forall 1 \leq j \leq J \end{gathered} \quad (22)$$

where $\tilde{\boldsymbol{\alpha}}_l = (\boldsymbol{\alpha}_l; \boldsymbol{q}_l)$ and $\mathcal{H}_l$ is defined in (21). The solution to (22) can be determined by any standard nonlinear programming techniques [14], e.g. active-set, cutting plane or interior point methods. Since our experiments are performed in the Matlab environment, a Matlab build-in function is employed to solve (22).

4 Experiments

In this section, we compare the proposed RADM with state-of-the-art classifier fusion algorithms, including Sum [3], IN [7], LPBoost [12], LP-B [8], RM [6], DN [7] and LCDM [10], in four different domains of recognition problems. In Sections 4.1 and 4.2, these combination methods are evaluated with the Digit [18] and Flower [19] databases, respectively. After that, the results for face recognition using CMU PIE [20] and FERET [21], and human action recognition using Weizmann [22] and KTH [23] databases, are reported in Sections 4.3 and 4.4, respectively. It is important to point out that the main objective of these experiments is to evaluate the performance of different classifier fusion methods, but not state-of-art digit, flower, face and human action recognition algorithms.

4.1 Digit Recognition

Multiple feature digit database [18] contains ten digits from 0 to 9, and 200 examples for each digit. Six features, namely Fourier coefficients, profile correlations, Karhunen-Love coefficients, pixel averages, Zernike moments and morphological features, are extracted [18]. In this experiment, we randomly select 20 samples of each digit for training and the rest for testing. Since the probabilities are hard to determine accurately due to the problem of limited training samples, we use SVM classifiers [24] and normalize the SVM outputs by the double sigmoid method [25] to approximate the probabilities. To select the best parameters, five-fold cross validation (CV) is performed. The parameter C introduced in the soft margin SVM is selected from $\{0.01, 0.1, 1, 10, 100, 1000\}$. The CV outputs of the SVMs are used to train the weights for classifier combination. The RADM parameters η and ε_0 in (22) are set as follows. If features are independent, the point-wise dependencies $\frac{\Pr(\boldsymbol{x}_{j1},\cdots,\boldsymbol{x}_{jM})}{\prod_{m=1}^{M}\Pr(\boldsymbol{x}_{jm})}$ for $1 \leq j \leq J$ are equal to 1. Since parameter η represents the lower bound of the point-wise dependencies, we set $\eta = 0.5 < 1$, so that RADM includes the independent case. On the other hand, parameter ε_0 in (22) must be a small number, so ε_0 is set as 0.01. Other parameters are selected from suitable sets as follows. Regularization parameters b, c are selected from $\{10^{-6}, \cdots, 10^{-1}, 1\}$, the variable order R is selected from $\{1, 2\}$ and the model order K is selected from one to the number of features. For LPBoost, LP-B and LCDM, the best parameter ν is selected from $\{0.05, 0.1, \ldots, 0.95\}$. This experiment has been repeated ten times.

The mean accuracies of the best single feature (BestFea) and different classifier combination methods are reported in the second column of Table 1. From Table 1, we can see that the proposed RADM obtains the highest recognition rate of 96.84% on this database. Moreover, the recognition accuracies of the classifier fusion methods are higher than that of the best single feature. This convince that the performance can be improved by combining classifiers, and RADM outperforms other fusion methods by better modeling dependency.

4.2 Flower Recognition

Oxford flowers database [19] contains 17 categories of flowers with 80 images per category. Seven features including shape, color, texture, HSV, HoG, SIFT internal, and SIFT boundary, are extracted using the methods reported in [19] [26]. We evaluate the proposed method using 17 × 40 images for training, 17 × 20 for validation and 17 × 20 for testing. The best parameters are selected by the

Table 1. Recognition accuracies (%) of different methods on all databases

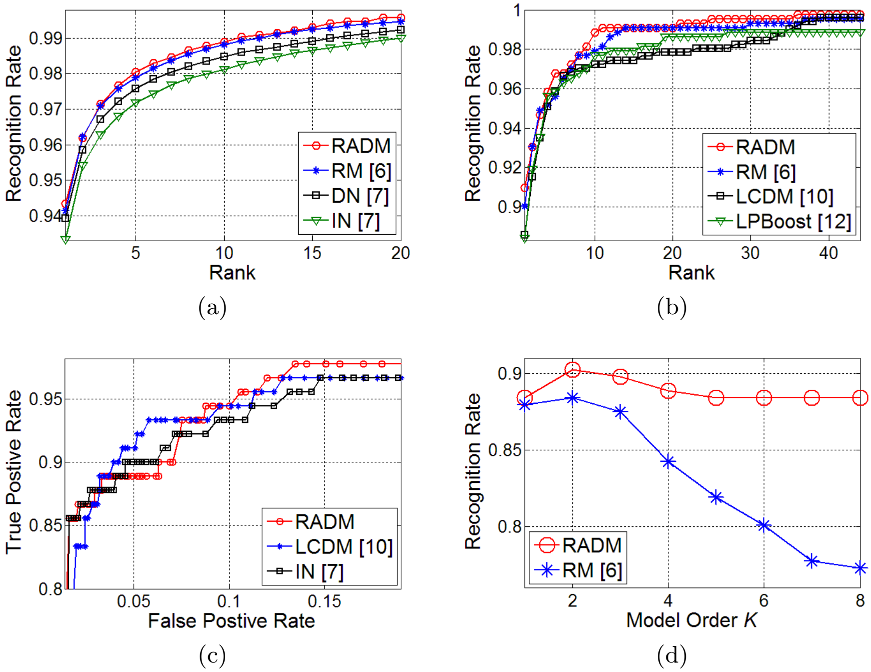
Dataset Method	Digit	Flower	CMU PIE	FERET	Weizmann	KTH
BestFea	94.77	70.39	88.87	83.33	82.22	78.70
Sum [3]	96.23	85.39	91.75	86.11	84.44	84.72
IN [7]	95.63	85.49	93.32	88.19	**85.56**	84.26
LPBoost [12]	96.41	82.74	92.14	88.43	83.33	83.33
LP-B [8]	96.57	85.49	92.00	87.65	84.44	85.19
RM [6]	96.71	85.39	94.14	90.05	84.44	88.43
DN [7]	94.93	84.22	93.91	87.73	84.44	83.80
LCDM [10]	96.79	86.27	93.01	88.81	**85.56**	85.19
RADM	**96.84**	**88.04**	**94.34**	**90.97**	**85.56**	**90.28**

validation set similar to the procedure in Digit database. Following the settings in [8], kernel SVMs are used in this experiment, and the kernel matrices are defined as $\exp\left(-d(x,x')/\lambda\right)$, where d is the distance and λ is the mean of pairwise distances. This experiment was repeated three times using the predefined splits of this database [19].

The third column in Table 1 shows the mean accuracies of different methods. From Table 1, we can see that RADM obtains an improvement of 2.55% and 1.77% over the classifier fusion methods with independent assumption and those without independent assumption, respectively. This indicates that modeling dependency without specific assumptions like those in DN and LCDM helps to improve the recognition performance.

4.3 Face Recognition

Two pubilicly available face databases, CMU PIE [20] and FERET [21], are used for classifier fusion experiments. CMU PIE face database contains 68 subjects with 41,368 images captured under varying pose, illumination and expression. We used 105 near frontal face images for each individual, randomly select six for training, four for validation and the rest for testing. In FERET database, we select 72 individuals with six near frontal face images per person under different face expressions. Six images for each individual are randomly separated into training, validation and testing sets with equal size, i.e. two images for each set. Four features, Eigenfaces [27], Fisherfaces [27], Laplacianfaces [2] and local binary patterns (LBP) [1] are extracted in both two databases. Inspired by the experiments in [1] [2] [27], parameters introduced from these features are determined as follows. The dimensions of Eigenfaces, Fisherfaces and Laplacianfaces are set as 100 and the number of nearest neighbors in Laplacianfaces is set to be 3, while the window size is set as 16 × 12 with $\mathrm{LBP}^{u2}_{8,2}$. SVM outputs of the training data are used to train the weights for classifier fusion methods. The best parameters are selected by the validation set similar to the procedure mentioned in Section 4.1. These experiments were repeated ten times on CMU PIE and three times on FERET database.



(a) (b)

(c) (d)

Fig. 1. (a) (b) CMC curves on CMU PIE and FERET face databases. (c) ROC curve on Weizmann database. (d) Recognition rates of RADM and RM with changed model order on KTH database.

The mean accuracies on these two databases are reported in the fourth and fifth columns of Table 1. Same conclusion can be drawn that RADM outperforms the other methods on both two face databases. While results in Table 1 only show the Rank-1 accuracies, CMC curves of the top four methods on CMU PIE and FERET databases are plotted in Fig. 1(a) and Fig. 1(b), respectively, for detailed comparison. It can be seen that RADM outperforms IN and DN on CMU PIE, LPBoost and LCDM on FERET database, and is slightly better than RM with different number of ranks. This indicates that RADM with dependency modeling gives the best performance for face recognition as well.

4.4 Human Action Recognition

In this section, we compare the classifier fusion methods on Weizmann [22] and KTH [23] human action databases. Weizmann database contains 93 videos from nine persons, each performing ten actions. Eight out of the nine persons in this database are used for training, and the remaining one is used for the evaluation. This is repeated nine times and the rates are averaged. On the other hand, there are 25 subjects performing six actions under four scenarios in KTH database. We follow the common setting [23] to separate the video set into training (8 persons), validation (8 persons), and testing (9 persons) sets. Eight features including intensity, intensity difference, HoF, HoG, HoF2D, HoG2D, HoF3D and HoG3D, are extracted from videos as reported in [10]. In these two databases, eight-fold CV is performed on the training data, and the CV outputs are used to train the weights for classifier fusion. The best parameters are selected by the CV outputs on Weizmann and validation set on KTH database, respectively, similar to the procedure mentioned in Section 4.1.

The last two columns in Table 1 show the recognition rates of different methods. We can see that RADM, LCDM and IN get the highest recognition rate of 85.56% on Weizmann, while RADM outperforms other methods on KTH database. We further compare the best three algorithms on Weizmann database by the ROC measurement. The ROC curves in Fig. 1(c) show that RADM gives better performance when the false positive rate is larger than 10%. And the areas under curves (AUC) are 0.8524 for RADM, 0.8472 for LCDM and 0.8424 for IN. This also convinces that the proposed RADM is better than other classifier fusion methods for human action recognition. At last, we compare RADM and RM with changed model order K on KTH database. From Fig. 1(d), we can see that RADM outperforms RM with different model orders and is less sensitive to model order changed. This is another advantage of the proposed method.

5 Conclusion

In this paper, we have designed and proposed a new framework for dependency modeling by analytical functions on posterior probabilities of each feature. It is shown that Product rule [3] (with independent assumption) and LCDM [10] (without independent assumption) can be unified by the proposed analytical dependency model (ADM). With the ADM, we give an equivalent condition to independent assumption from the properties of marginal distributions, and show that ADM can model dependency. Since ADM may contain infinite number of undetermined coefficients, a reduced form is proposed based on the convergent properties of analytical functions. At last, the optimal Reduced Analytical Dependency Model (RADM) is learned by a modified least square regularization problem, which aims at approximating posterior probabilities such that all training samples are correctly classified. Experimental results show that RADM outperforms existing classifier fusion methods on Digit, Flower, Face and Human Action databases. This indicates that RADM can better model dependency and help to improve the performance in many recognition problems.

Acknowledgments. This project was partially supported by the Science Faculty Research Grant of Hong Kong Baptist University, National Natural Science Foundation of China Research Grants 61128009 and 61172136, and NSFC-GuangDong Research Grant U0835005.

References

1. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Recognition with Local Binary Patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
2. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using Laplacianfaces. TPAMI 27, 328–340 (2005)

3. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. TPAMI 20, 226–239 (1998)
4. Ueda, N.: Optimal linear combination of neural networks for improving classification performance. TPAMI 22, 207–215 (2000)
5. Prabhakar, S., Jain, A.K.: Decision-level fusion in fingerprint verification. Pattern Recognition 35, 861–874 (2002)
6. Toh, K.A., Yau, W.Y., Jiang, X.: A reduced multivariate polynomial model for multimodal biometrics and classifiers fusion. IEEE Transactions on Circuits and Systems for Video Technology 14, 224–233 (2004)
7. Terrades, O.R., Valveny, E., Tabbone, S.: Optimal classifier fusion in a non-bayesian probabilistic framework. TPAMI 31, 1630–1644 (2009)
8. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV, pp. 221–228 (2009)
9. Scheirer, W., Rocha, A., Micheals, R., Boult, T.: Robust Fusion: Extreme Value Theory for Recognition Score Normalization. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 481–495. Springer, Heidelberg (2010)
10. Ma, A.J., Yuen, P.C.: Linear dependency modeling for feature fusion. In: ICCV, pp. 2041–2048 (2011)
11. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall (1986)
12. Demiriz, A., Bennett, K.P., Shawe-Taylor, J.: Linear programming boosting via column generation. JMLR 46, 225–254 (2002)
13. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1. Wiley (1968)
14. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming. Springer (2008)
15. Krantz, S.G., Parks, H.R.: A Primer of Real Analytic Functions. Birkhäuser (2002)
16. Rudin, W.: Principles of mathematical analysis. McGraw-Hill (1976)
17. Rifkin, R., Yeo, G., Poggio, T.: Regularized least squares classification. NATO Science Series III: Computer and Systems Sciences 190, 131–153 (2003)
18. Breukelen, M.V., Duin, R.P.W., Tax, D.M.J., Hartog, J.E.D.: Handwritten digit recognition by combined classifiers. Kybernetika 34, 381–386 (1998)
19. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: CVPR (2006)
20. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression database. TPAMI 25, 1615–1618 (2003)
21. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. TPAMI 22, 1090–1104 (2000)
22. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. TPAMI 29, 2247–2253 (2007)
23. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: ICPR (2004)
24. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: SVM and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France (2005)
25. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern Recognition 38, 2270–2285 (2005)
26. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: ICVGIP (2008)
27. Peter, N., Belhumeur, J.P.H., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. TPAMI 19, 711–720 (1997)