Neurocomputing 123 (2014) 100–109

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Online heterogeneous feature fusion machines for visual recognition

Shuangping Huang [a], Lianwen Jin [b,*], Yuan Fang [c], Xiaoxin Wei [b]

[a] College of Engineering, South China Agricultural University, Guangzhou, PR China
[b] School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, PR China
[c] College of Computer & Information Science, Northeastern University, Boston, USA

ARTICLE INFO

Article history:
Received 25 November 2012
Received in revised form
13 April 2013
Accepted 19 June 2013
Communicated by D.S. Huang
Available online 2 July 2013

Keywords:
Online optimization
Heterogeneous feature fusion
Visual recognition
Group LASSO
Multiple kernels learning

ABSTRACT

Heterogeneous Feature Fusion Machines (HFFM) is a kernel based logistic regression model that effectively fuses multiple features for visual recognition tasks. However, the batch mode solution for HFFM, 'Block Coordinate Gradient Descent' (BCGD) has the same low efficiency and poor scalability as the most batch algorithms do. In this paper, we describe a newly developed online learning algorithm in multiple Reproducing Kernel Hilbert Spaces for solving HFFM model. This new algorithm is called OLHFFM, i.e. Online HFFM. OLHFFM is novel combination of kernel-based learning technique with dual averaging gradient descent methods. In addition, group LASSO regularization technique is used in OLHFFM for finding important explanatory coefficients that are related to support samples in group manner. The effectiveness of OLHFFM has been demonstrated by a number of experiments that were conducted on public event, object dataset, as well as on large scale handwritten digital dataset. Using the OLHFFM approach, we have achieved almost equivalent recognition performance to that using batch-mode approach. Experiments conducted on both MIT Caltech-6 and challenging VOC2011 TrainVal object datasets show that OLHFFM is superior in performance to kernel based online learning approaches such as ILK or NORMA. In addition, the classification performance of OLHFFM approach as demonstrated by the experiments conducted on large scale MNIST dataset is comparable to or better than that of the current state-of-the-art online multiple kernel learning approaches such as OM-2, UFO-MKL, OMCL and OMKL. Extensive experiments on visual data classification demonstrate the effectiveness and robustness of the new OLHFFM approach.

1. Introduction

Impressive progress has been made in visual recognition field recently [1–25,38,41,44,47]. An important part of recent work has been focusing on a range of advanced image descriptors, for example, SIFT (Scale Invariant Feature Transform) [2], GIST [3], Histogram of Oriented Gradients [4], Local Binary Pattern (LBP) [5], CTM (Color Texture moment) [6], shape context [7] and so on. Different features describe different aspects of the visual characteristics and cover different visual recognition cues such as appearance, texture, shape and color. They are complementary to each other. For more and more complex visual recognition tasks, one cannot just use single type feature since it does not provide enough discriminative information. For this reason, combination of heterogeneous features is gaining popularity lately for more complex visual recognition tasks [8–21,47].

Technical challenges still exist for fusing multiple features in general way. A large number of publications can be found in this subject, even though they could be under different names such as "multiple view learning" or "multiple features/heterogeneous feature fusion". These publications fall into the following categories according to the intrinsic nature of their formulations: (1) multiple feature fusion by projection or subspace learning [9,10]; (2) multiple feature fusion by combining different kernels that corresponds to different measures of similarity for different representations [11–15]; (3) multiple feature fusion by means of spectral embedding, e.g. multiview (MSE) [16] or distributed (DSE) [17]; and (4) methods based on data graph wherein convexly combining the graph Laplacians on different views [18,19]. Besides, there are some case-by-case algorithmic instantiation in solving specific real-world problems, for example, Co-LapSVM and Co-LapRLS within co-regularization framework for semi-supervised learning [20]; m-SNE based on stochastic neighbor embedding [21] etc. Multiple Kernel Learning (MKL) based models are among the most popular ones because it is the most reasonable approach for combining multiple information sources. MKL combines different kernels for different features by a weighted summation. The weight for each feature does not depend on one certain sample

* Corresponding author. Tel.: +86 20 87113540.
E-mail addresses: shuangping.huang@gmail.com (S. Huang), lianwen.jin@gmail.com, eelwjin@scut.edu.cn (L. Jin).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.06.004

and remains the same across all the samples [24]. Recently, a new HFFM model gains popularity. In this new HFFM model, the weights of kernels vary from sample to sample. This leads to nonlinear fusion of the multiple kernels functions. It makes kernel combination more feasible and thus promotes its fusion capability [24,47].

For visual recognition tasks, batch mode solution has been used for heterogeneous feature fusion. It is well known that batch solution approach has the limit of poor scalability, low efficiency, and high cost [24,35]. It even becomes impractical to use batch solution approach when one has to handle millions of image samples. As a result, online learning algorithms have gained popularity for their high efficiencies in large-scale data analysis [26-33,40,50]. Another advantage of online algorithm is the ability to "include human in the loop" with robotic vision.

In this paper, we describe a novel online algorithm called OLHFFM in multiple Reproducing Kernel Hilbert Spaces that combines group LASSO sparse method and dual averaging sub-gradient learning technique. This online algorithm is used to solve HFFM model efficiently and it can be used for a wide range of visual recognition tasks such as event recognition, object categorization and so on. Different than standard online MKL, the solution of HFFM tends to depend on a subset of low-noise samples. Group LASSO is used to select explanatory samples and remove noisy samples in HFFM model for the classifying function. In our work we demonstrated the feasibility of implementing non-linear multiple kernel fusion technique in an online mode.

The remainder of this paper is organized as follows. Section 2 reviews the related online learning works especially that focuses on online solutions for standard MKL model. Section 3 states the problem of HFFM models and formulizes the online HFFM model solution; Section 3 presents some analytical and comparative experiments on a variety of visual recognition task delivered on some publicly available benchmark datasets and show some findings. Section 4 concludes this study with future work.

2. Related work

Some examples of online algorithms that are used in linearly separable cases include Rosenblatt's Perceptron [26], FOBOS method developed by Duchi and Singer [27], RDA (Regularized Dual Averaging) [28,50] and DA-GL (Dual Averaging-Group LASSO) [29]. Among these, FOBOS can be considered as a general framework for stochastic gradient with arbitrary regularization. It alleviates the problems of non-differentiability in cases such as ℓ_1-regularization by taking analytical minimization steps interleaved with sub-gradient steps. RDA (Regularized Dual Averaging) [28] method is proposed by Lin Xiao for solving regularized learning problems under online setting as FOBOS does. The uniqueness of RDA is that it updates model parameter vector by solving a simple optimization problem on each round that involves average of all the past sub-gradients of the loss functions and the whole regularization term. That is the main difference between RDA and FOBOS. In essence, RDA adjusts the learning variables not only using information from the single coming example but also from sub-gradients of loss functions as to past examples. However, only simple LASSO regularization is considered in RDA for sparsity. DA-GL [29] inherits the idea of RDA and it is extended for solving a group LASSO regularized optimization problem in the original feature space. Moreover, various variants of group LASSO such as sparse group LASSO [21], group LASSO with overlap and graph LASSO [22] can be adopted as online algorithms' regularization item for finding important explanatory variable group. It is noted that attributes in the feature space form a group here. That is to say, d-dimension feature vector is divided into G groups with d_g (the number of attributes in g-th group). The number d_g is usually assumed greater than 1. When a group is sparsified by means of various group LASSO methods, corresponding feature attributes are set to zero in the model.

For linearly inseparable data analysis another family of online algorithms with kernel integration is used. Some examples of single-kernel based online algorithms include NORMA (Naïve Online R-reg Minimization Algorithm) [30], ILK (Implicit online Learning with Kernels) [31] and so on. Both NORMA and ILK perform gradient descent with respect to ℓ_2-regularized instantaneous risk in Reproducing Kernel Hilbert Spaces (RKHS). The main difference between them is that NORMA implements explicit parameter updates and ILK implements implicit ones. However, both of them yield no sparse solutions since they involves ℓ_2-norm in RKHS which produces only soft shrinkage on hypothesis. In practice, it is often desirable to seek LASSO [18] or group LASSO [19] with sparsity at group level or individual level. Some state-of-the-art multiple kernel based online algorithm examples including UFO-MKL [14], OBSCURE [33], and OM-2 [11] adopt feasible regularization technique to obtain tunable sparsity or make optimization problem easier. Among these, UFO-MKL mixes elements of group p-norm and LASSO, i.e. forming elastic net kind of regularization which separately provides an easy optimization problem and induces feasible levels of sparsity in the domain of the kernels. Stochastic gradient descent and mirror descent are combined and used to solve this elastic net kind regularized MKL problem. The solution of the optimization problem will lead to a selection a subset of the F kernels. OBSCURE is proposed to solve p-norm version of the standard MKL model. It highlights two-stage optimization method. The first stage is an online initialization procedure that determines quickly the region of the space where the optimal solution lives. The second one refines the solution found by the first stage. OM-2 uses the 'Follow the Regularized Leader' [48,49] framework to solve group p-norm regularized MKL problem. The other two algorithms including OMKL [15] and OMCL [32] focus not on regularization but on decomposing MKL solution into two separate tasks. OMKL uses deterministic or stochastic approaches to combine binary predictions or real-valued outputs from multiple kernel classifiers. The deterministic approach updates all kernel classifiers for every misclassified example, while the stochastic approach chooses a classifier(s) randomly for updating according to some sampling strategies. Different setup, i.e. deterministic or stochastic, binary predictions or real-valued outputs, forms OMKL series algorithms. OMCL is a wrapper algorithm using a two-layer structure, which can use most of the known online learning methods as base algorithms. However, all of these algorithms are part of the standard Multiple Kernel Learning (MKL) family. That means that they aim to obtain multiple kernels classifier and their linear combinations from a pool of given kernels in an online fashion. Moreover, the weights *wm* for the *mth* kernel remain the same across all the samples. We emphasize that although a number of approaches have been proposed to solve the optimization problem related to MKL, little work has been done to address online HFFM learning. To the best of our knowledge, this is the first theoretic study that addresses the online HFFM problem.

3. Online HFFM algorithm

3.1. Preliminaries

Before presenting OLHFFM learning method, we first describe briefly HFFM [24,47] and introduce some basic notations for classification. Assume $\{\boldsymbol{x_i}, y_i\}$ is an input–output pair. Here $y_i \in \{1, 0\}$ is the label of a sample for binary classification problem and

$\boldsymbol{x_i}=[\boldsymbol{x_i^1},\ldots,\boldsymbol{x_i^{\mathscr{M}}}]$ is the corresponding measurement of $\mathscr{M}$ features, where each $\boldsymbol{x_i^j}(j=1,2,\ldots,\mathscr{M})$is a feature vector that describes a visual characteristic of an image. For each feature j, the similarity metric between two samples is represented by $k^j(\boldsymbol{x},\boldsymbol{x}_i)$ which is in essence a kernel function.

The HFFM model is formulated as

$$\mathscr{F}(\boldsymbol{x})=f^0+\sum_i\sum_{j=1}^{\mathscr{M}}f^{i,j}k^j(\boldsymbol{x},\boldsymbol{x}_i) \tag{1}$$

where f^0 is bias item and $f^{i,j}$ is coefficient of $k^j(\boldsymbol{x},\boldsymbol{x}_i)$. To solve $\mathscr{F}(\boldsymbol{x})$, it minimizes a composite objective function

$$C=-L(\boldsymbol{f})+\lambda\sum_i\sqrt{d_i}\|\boldsymbol{f}^{i,\cdot}\|_2 \tag{2}$$

where the first term is an empirical loss function over the whole training example set and $f^{i,1}=[f^{i,1},f^{i,2},\ldots,f^{i,\mathscr{M}}]$ is a coefficients vector related with the i-th sample. It is the negative log-likelihood from a logistic regression defined by

$$-L(\boldsymbol{f})=-\sum_i\log\frac{exp(y_i\mathscr{F}(\boldsymbol{x}_i))}{1+exp(\mathscr{F}(\boldsymbol{x}_i))} \tag{3}$$

The second term is the regularization term of group LASSO [34] where $\|.\|_2$ denotes ℓ_2- norm. λ is a tunable parameter that stands for the tradeoff between logistic loss and group LASSO regularization. $\sqrt{d_i}$ is the degree of freedom, here it is equal to$\sqrt{\mathscr{M}}$ and $\boldsymbol{f}^{i,\cdot}=[f^{i,1},f^{i,2},\ldots f^{i,\mathscr{M}}]$. Group LASSO is used to produce group sparsity. From the statement, the HFFM model aims to learn multiple functions in $\mathscr{M}$ kernel space jointly. However, standard MKL aims to learn a single function in the space of $\mathscr{H}_{\mathcal{K}u}$, where$\mathcal{K}u(\cdot,\cdot)=\sum_{j=1}^{\mathscr{M}}u_j\mathcal{K}_j(\cdot,\cdot)$, $u\in\Delta$, Δ denotes a simplex.

The block Co-ordinate Gradient Descent (BCGD) [24] method is used to solve the model. From Eqs. (2) and (3), we understand that batch algorithm has to scan all the training samples at each iteration. This means significant computational cost. This problem can be solved with online learning approach that is proposed in this work. In the online learning approach, the learning variables are adjusted using simple calculations based on a single example at a time and it only needs to scan through training examples.

3.2. Online learning

According to online learning framework, a set of hypotheses $\boldsymbol{f}=\{\boldsymbol{f}_{(1)},\boldsymbol{f}_{(2)},\ldots\}$ is desired to produce from the learning process as training examples become available one by one. Here $\boldsymbol{f}_{(1)}$ is some arbitrary initial hypothesis, i.e. $\boldsymbol{f}_{(1)}=0$ which means that the algorithm starts with the zero hypothesis and $\boldsymbol{f}_{(i)}$ for $i>1$ is the hypothesis chosen after seeing the $(i-1)$th example. From the description of HFFM model, dimension of parameter vector $\boldsymbol{f}_{(t)}$ is $t\times\mathscr{M}$. For the clarity of our online algorithm formulation we rewrite the long parameter vector$\boldsymbol{f}_{(t)}$ in matrix form

$$\boldsymbol{f}_{(t)}=\begin{bmatrix}f_{(t)}^{1,1}&\cdots&f_{(t)}^{1,j}&\cdots&f_{(t)}^{1,\mathscr{M}}\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ f_{(t)}^{i,1}&\cdots&f_{(t)}^{i,j}&\cdots&f_{(t)}^{i,\mathscr{M}}\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ f_{(t)}^{t,1}&\cdots&f_{(t)}^{t,j}&\cdots&f_{(t)}^{t,\mathscr{M}}\end{bmatrix}_{t\times\mathscr{M}} \tag{4}$$

Each row in the Eq. (4) represents the coefficients related with a sample and each column stands for the coefficients related with one certain kernel. The expression highlights two different controlling indices i,j that represent sample sequence and kernel sequence number, respectively. Here we can further simplify the Eq. (1) as

$$\mathscr{F}(\boldsymbol{x})=f^0+\sum_i\boldsymbol{f}^{i,\cdot}\times\mathcal{K}^T(\boldsymbol{x},\boldsymbol{x}_i) \tag{5}$$

Or,

$$\mathscr{F}(\boldsymbol{x})=f^0+\sum_j(\boldsymbol{f}^{\cdot,j})^T\times\boldsymbol{k}^j(\cdot,\boldsymbol{x}) \tag{6}$$

In Eq. (5)

$$\mathcal{K}(\boldsymbol{x},\boldsymbol{x}_i)=[k^1(\boldsymbol{x},\boldsymbol{x}_i),k^2(\boldsymbol{x},\boldsymbol{x}_i),\ldots,k^{\mathscr{M}}(\boldsymbol{x},\boldsymbol{x}_i)] \tag{7}$$

In Eq. (6)

$$(\boldsymbol{f}^{\cdot,j})=[f^{1,j},f^{2,j},\ldots f^{t,j}]^T \tag{8}$$

and

$$\boldsymbol{k}^j(\cdot,\boldsymbol{x})=[k^j(\boldsymbol{x}_1,\boldsymbol{x}),k^j(\boldsymbol{x}_2,\boldsymbol{x}),\ldots,k^j(\boldsymbol{x}_t,\boldsymbol{x})]^T \tag{9}$$

In fact, all the corresponding kernel function values used in Eqs. (5) and (6) can be combined into multiple kernel matrix as

$$\mathcal{K}(\cdot,\boldsymbol{x})=\begin{bmatrix}k^1(\boldsymbol{x}_1,\boldsymbol{x})&\cdots&k^j(\boldsymbol{x}_1,\boldsymbol{x})&\cdots&k^{\mathscr{M}}(\boldsymbol{x}_1,\boldsymbol{x})\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ k^1(\boldsymbol{x}_i,\boldsymbol{x})&\cdots&k^j(\boldsymbol{x}_i,\boldsymbol{x})&\cdots&k^{\mathscr{M}}(\boldsymbol{x}_i,\boldsymbol{x})\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ k^1(\boldsymbol{x}_t,\boldsymbol{x})&\cdots&k^j(\boldsymbol{x}_t,\boldsymbol{x})&\cdots&k^{\mathscr{M}}(\boldsymbol{x}_t,\boldsymbol{x})\end{bmatrix}_{t\times\mathscr{M}} \tag{10}$$

Since we are interested in online algorithms that deal with one example at a time, we also define an instantaneous logistic loss similar to HFFM by

$$\ell_t(\mathscr{F}(\boldsymbol{x}_t),y_t)=-log\frac{exp(y_t\mathscr{F}_{(t)}(\boldsymbol{x}_t))}{1+exp(\mathscr{F}_{(t)}(\boldsymbol{x}_t))} \tag{11}$$

For simplicity, we denote the sub-gradient as $g_{(t)}=(\partial\ell_{(t)}/\partial\boldsymbol{f}_{(t)})$ ($g_{(t)}^{i,j}=(\partial\ell_t/\partial f_{(t)}^{i,j})$) and it is also given matrix expansion form as

$$g_{(t)}=\begin{bmatrix}g_{(t)}^{1,1}&\cdots&g_{(t)}^{1,j}&\cdots&g_{(t)}^{1,\mathscr{M}}\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ g_{(t)}^{i,1}&\cdots&g_{(t)}^{i,j}&\cdots&g_{(t)}^{i,\mathscr{M}}\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ g_{(t)}^{t,1}&\cdots&g_{(t)}^{t,j}&\cdots&g_{(t)}^{t,\mathscr{M}}\end{bmatrix}=\begin{bmatrix}\frac{\partial\ell_t}{\partial f_{(t)}^{1,1}}&\cdots&\frac{\partial\ell_t}{\partial f_{(t)}^{1,j}}&\cdots&\frac{\partial\ell_t}{\partial f_{(t)}^{1,\mathscr{M}}}\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ \frac{\partial\ell_t}{\partial f_{(t)}^{i,1}}&\cdots&\frac{\partial\ell_t}{\partial f_{(t)}^{i,j}}&\cdots&\frac{\partial\ell_t}{\partial f_{(t)}^{i,\mathscr{M}}}\\ \vdots&\ddots&\vdots&\ddots&\vdots\\ \frac{\partial\ell_t}{\partial f_{(t)}^{t,1}}&\cdots&\frac{\partial\ell_t}{\partial f_{(t)}^{t,j}}&\cdots&\frac{\partial\ell_t}{\partial f_{(t)}^{t,\mathscr{M}}}\end{bmatrix}_{t\times M} \tag{12}$$

From Eq. (11)

$$g_{(t)}=\frac{\partial\ell_{(t)}(\mathscr{F}(\boldsymbol{x}_t),y_t)}{\partial\boldsymbol{f}_{(t)}}=\ell'(\mathscr{F}(\boldsymbol{x}_t),y_t)\times\mathcal{K}(\cdot,\boldsymbol{x}_t) \tag{13}$$

where

$$\ell'(\mathscr{F}(\boldsymbol{x}_t),y_t)=\frac{exp(\mathscr{F}(\boldsymbol{x}_t))}{1+exp(\mathscr{F}(\boldsymbol{x}_t))}-y_t=u_t \tag{14}$$

According to the framework of Dual Averaging Gradient method [28,50], the learning variables are adjusted by solving a simple optimization problem that involves the running average of all the past sub-gradients of instantaneous loss functions ℓ_t and the whole regularization term. We first take a look at the calculation of average gradient matrix which accounts for accumulated effect from all the past rounds and is denoted as $\overline{g}_{(t)}$. It is defined as

$$\overline{g}_{(t)}=\frac{t-1}{t}\overline{g}_{(t-1)}+\frac{1}{t}g_{(t)} \tag{15}$$

From Eq. (12), sub-gradient matrix $g_{(i)}$ has different rows. To calculate average sub-gradient, we need to expand g matrix for the consistency of the matrix structure. Therefore, it is critical to design the right added values for these g matrice and make them meaningful in practice. We define the adding rule as: at each round t, $u_i\mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_t)$ is added to $g_{(i)}(1\le i\le t-1)$ as its last row $g_{(i)}^{t,\cdot}$. So there are total of $(t-i)$ rows that are added according to the order of coming samples. In essence, the operation rule implies effect on current average gradient from all the previous samples. In other words, it can also be considered as accumulation of past effect or utilization of the internal knowledge involving past samples.

Based on the above mentioned design of added value of $g_{(i)}$, one can compute $\overline{g}_{(t)}$ as

$$\overline{g}_{(t)} = \frac{1}{t}\sum_{p=1}^{t} u_p \times \mathcal{K}(\cdot, \boldsymbol{x}_p) \tag{16}$$

or compute it according to Eq. (15) along the sequence of observations.

With the average gradient information, the new OLHFFM algorithm solves the following minimization problem at each round:

$$\boldsymbol{f}_{(t+1)} = \operatorname*{argmin}_{\boldsymbol{f}} \left\{ \sum_{j}^{\mathscr{M}} (\boldsymbol{f}^{\cdot,j})^T \ \overline{g}_{(t)}^{\cdot,j} + \mathscr{R}(\boldsymbol{f}) + \frac{\gamma}{\sqrt{t}} h(\boldsymbol{f}) \right\} \tag{17}$$

Where, $\sum_j^{\mathscr{M}} (\boldsymbol{f}^{\cdot,j}) \overline{g}_{(t)}^{\cdot,j}$ is the item associated with Bregman Divergence [36]; $\mathscr{R}(\boldsymbol{f})$ is group LASSO regularization item defined on hypothesis $\boldsymbol{f}$. Here we can write it as

$$\mathscr{R}(\boldsymbol{f}) = \lambda \sum_{i=1}^{t} \sqrt{d_i}\, \|\boldsymbol{f}^{i,\cdot}\|_{\mathscr{H}} \tag{18}$$

where the degree of freedom d_i in this scenario is the number of kernels which is equal to $\mathscr{M}$. $h(\boldsymbol{f})$ is an auxiliary strongly convex function. Let $h(\boldsymbol{f}) = (1/2)\boldsymbol{f}_{\mathscr{H}}^2$ for the simple optimal solution at each iteration, it makes us find a closed-form solution as

$$\boldsymbol{f}_{(t+1)}^{i,\cdot} = \begin{cases} 0 & , 1 - \frac{\lambda\sqrt{\mathscr{M}}}{\|\overline{g}_t^{i,\cdot}\|} \le 0 \\ -\frac{\sqrt{t}}{\gamma}\left(1 - \frac{\lambda\sqrt{\mathscr{M}}}{\|\overline{g}_t^{i,\cdot}\|}\right)\overline{g}_t^{i,\cdot} & , \text{others} \end{cases} \tag{19}$$

Since $\mathscr{F}(\boldsymbol{x}) = f^0 + \sum (\boldsymbol{f}^{\cdot,j})^T \times \mathcal{K}^j(\cdot, \boldsymbol{x})$, we rewrite each $\boldsymbol{f}_{(t)}^{\cdot,j}$ as a kernel expansion in terms of Representer Theorem [37].

$$\boldsymbol{f}_{(t)}^{\cdot,j} = \sum_{\mathcal{P}=1}^{t-1} \alpha_{(t)}^{\mathcal{P},j} k^j(\cdot, \boldsymbol{x}_{\mathcal{P}}) \ \ (j = 1, 2, 3, \ldots, \mathscr{M}) \tag{20}$$

where, $\boldsymbol{\alpha}_{(t)}^{\cdot,j}$ is the jth kernel expansion coefficients group. From Eq. (20), each kernel expansion $\boldsymbol{f}_{(t)}^{\cdot,j}$ forms a column of $\boldsymbol{f}_{(t)}$ matrix. What makes this model advantageous is the group LASSO that reduces the model complexity. It is desirable to maintain the group LASSO for sparsity in the online setting in order to improve model's performance. However, one cannot use Eq. (20) to sparse at group level. Since column-structure of $\boldsymbol{\alpha}_{(t)} = [\boldsymbol{\alpha}_{(t)}^{\cdot,1}, \boldsymbol{\alpha}_{(t)}^{\cdot,2}, \ldots, \boldsymbol{\alpha}_{(t)}^{\cdot,\mathscr{M}}]$ is not based on 'group structure' nor does it benefits group sparsity, we break $\boldsymbol{\alpha}_{(t)}$ into row-structure where each row corresponds to a group that implies the correspondence among all the elements within the same row. We then rewrite $\boldsymbol{f}_{(t)}$ as a kernel expansion based on 'group structure' with a zeros initial hypothesis $\boldsymbol{f}_1 = 0$, for $i = 1, 2, .., t$

$$\boldsymbol{f}_{(t)}^{i,\cdot} = \begin{cases} 0 & , t = 1 \\ \sum_{p=1}^{t-1} c_{(t-1)}^{i} \ll \boldsymbol{\theta}_{(t-1)}^{p,\cdot} K(x_i, x_p) \gg & , t > 1 \end{cases} \tag{21}$$

where

$$\boldsymbol{\theta}_{(t-1)}^{p,\cdot} = [\theta_{(t-1)}^{p,1}, \theta_{(t-1)}^{p,2}, \ldots, \theta_{(t-1)}^{p,\mathscr{M}}] \tag{22}$$

$c_{(t-1)}^{i}$ is called Group Coefficient here. It implies the correspondence of expansion coefficients within group and reveals the i th group sparsity condition threshold value. $\ll v_1, v_2 \gg$ denotes an element-wise product operation between any two vectors : $v_1 = [v_{11}, v_{12}, \ldots, v_{1d}]$ and $v_2 = [v_{21}, v_{22}, \ldots, v_{2d}]$, i.e.

$$v = \ll v_1, v_2 \gg = [v_{11} \times v_{21}, \ldots, v_{1d} \times v_{2d}]$$

Then it follows from (19) and (21) that $1 \le i \le t-1$, Let $[v]_+$ denote $\max\{0, v\}$

$$c_{(t)}^{i} = \left[1 - \frac{\lambda\sqrt{\mathscr{M}}}{\frac{1}{t}\sum_{p=1}^{t} u_p \times \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_p)}\right]_+ \tag{23}$$

If $c_{(t)}^{i} \le 0$, then ith row of the hypothesis $\boldsymbol{f}$ will be set to $\mathbf{0}$ for sparsity, otherwise it is updated using (19). At the same time, we need to update θ_t according to

$$\theta_{(t)}^{p,\cdot} = \begin{cases} -\frac{\sqrt{t-1}}{\gamma\sqrt{t}}\theta_{(t-1)}^{p,\cdot} & , \text{for } p = 1, 2, \ldots, t-1 \\ -\frac{1}{\gamma\sqrt{t}} u_t & , \text{for } p = t \end{cases} \tag{24}$$

Now we have to consider the calculation of the bias f^0 of HFFM. Here, since the bias is not regularized, it can be calculated by

$$f_{(t+1)}^0 = \arg\min_{f^0} \left\{ \overline{b}_{(t)} f^0 + \frac{\gamma}{2\sqrt{t}} (f^0)^2 \right\} = -\frac{\sqrt{t}}{\gamma} \tag{25}$$

where

$$\overline{b}_{(t)} = \frac{\partial \ell_{(t)}}{\partial f_t^0}$$

To summarize, the OLHFFM algorithm is outlined in Algorithm 1.

Algorithm 1. Online learning algorithm for HFFM

Initialization:
$\overline{g}_{(0)} = 0; \quad \theta_{(0)} = 0; \quad c_{(0)} = 0$

for t=1,2,3.... **do**
initialize $f^{i,\cdot}$ using (21)
given the instantaneous loss function ℓ_t , compute the u_t using (14)
update the average sub-gradient value
$\overline{g}_{(t)}^{i,\cdot} = \frac{1}{t}\sum_{p=1}^{t} u_p \tilde{n} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_p)$

calculate sparse threshold value $c_{(t)}^{i}$using (23), then Update
$\boldsymbol{f}_{(t+1)}$ using (19)
update $f_{(t+1)}^0 = -\frac{\sqrt{t}}{\gamma}\overline{b}_{(t)}$
update $\theta_{(t)}$ using (24)
end for.

After the parameters matrix f is learned, we obtain the classification function $\mathscr{F}(x)$ using Eqs. (5) or (6). The estimate of the posterior probability can be computed as

$$p = \frac{e^{\mathscr{F}(x)}}{1 + e^{\mathscr{F}(x)}} \tag{26}$$

which measures how likely a testing sample x belongs to a class and hence find the most possible class label.

From the Algorithm 1, it is noted that the time spent by the OLHFFM is dominated by line 2 in each iteration, which requires a complexity of $O(t.\mathscr{M})$ for the worst case. $\mathscr{M}$ is the number of kernels and t is the number of past samples. This complexity is common to other state-of-the-art online learning algorithms like OBSCURE, OM-2 and UFO-MKL.

3.3. Convergence rate analysis

In this section we provide a theoretical guarantee analysis for the convergence rate of OLHFFM algorithm to the optimal fixed solution. Suppose that f^* is the optimal fixed matrix which satisfies $hf^* \le D^2$ for some $D > 0$ and there exists a constant L which satisfied $\overline{g}_{T*}^2 \le L^2$. We define the average regret with respect to the optimal hypothesis f^* as

$$\overline{R}_t = \frac{\mathscr{R}_t}{t} \triangleq \frac{\sum_i^t \left\{ \ell_i(f_{(i)}) + \mathscr{R}(f_{(i)}) + \frac{\gamma}{\sqrt{i}} h(f_{(i)}) \right\} - \sum_i^t \left\{ \ell_i(f^*) + \mathscr{R}(f^*) + \frac{\gamma}{\sqrt{i}} h(f^*) \right\}}{t} \tag{27}$$

where

$$\sum_{i}^{t}\left\{\ell_i(f_{(i)})+\mathcal{R}(f_{(i)})+\frac{\gamma}{\sqrt{i}}h(f_{(i)})\right\} \quad and \quad \sum_{i}^{t}\left\{\ell_i(f^*)+\mathcal{R}(f^*)+\frac{\gamma}{\sqrt{i}}h(f^*)\right\}$$

stand for cumulative loss the OLHFFM suffered along its run and the cumulative loss of the optimal fixed hypothesis.

Using Theorem 2 in paper [29], we know that the average regret is upper bounded by

$$\frac{\left(\gamma\sqrt{t}D^2+\frac{L^2}{2\gamma}\sum_{i=1}^{t}\frac{1}{\sqrt{i}}\right)}{t} \tag{28}$$

A detailed proof can be found in [50]. Based on the integral $\int_{x=1}^{t}(1/\sqrt{x})=2\sqrt{t}-2$, we obtain the following inequality

$$\overline{R}_t\le\frac{\gamma D^2+\frac{L^2}{\gamma}}{\sqrt{t}} \tag{29}$$

when $\gamma=L/D$ and this leads to the average regret bound as

$$\overline{R}_t\le\frac{2DL}{\sqrt{t}} \tag{30}$$

Hence, the HFFM algorithm can achieve an optimal $O(1/\sqrt{t})$ convergence rate from the point of the regret bound. Furthermore, the sequence of primal variables are bounded by

$$\frac{1}{2}\|f_{(t-1)}-f^*\|^2\le 2D^2-\frac{D\sqrt{t}}{L}\overline{R}_t \tag{31}$$

Eq. (31) shows the bound for the difference between the learned weight and the optimal weight.

4. Experimental evaluation

We conducted three series of experiments for studying the behavior of OLHFFM algorithm in terms of classification performance, scalability, sparseness. We also made extensive comparison between OLHFFM and some 'state-of-the-art' work in the same category. The experiments are implemented on event, object and handwritten digit dataset. For evaluating binary classifier model, we use Average Precision (AP) value as performance index.

4.1. Evaluation on event recognition

We compare OLHFFM with batch BCGD algorithm in this series of experiments. They are conducted on two public event recognition datasets: the Princeton sports event dataset [38] and Jain's Flickr sports event dataset [39]. For simplicity, they are called as dataset A and B thereafter, respectively. There are 8 sports categories in dataset A: bocce, croquet, polo, rowing, snowboarding, badminton, sailing, and rock climbing. The total number of image is 1200 and each category varies from 137 to 250. Dataset B contains 2449 Flickr images and it covers five popular American sports: baseball, basketball, football, soccer, and tennis.

We follow the same experimental protocol as [24] for direct comparison: 70 randomly selected images from each event class are used for training, 60 of the remaining images are used for testing on dataset A; 50% images are selected randomly for training and the remaining 50% is used for testing on dataset B.

4.1.1. Comparison with batch-mode HFFM

To compare batch and online modes, we follow the same experiment evaluation procedure as in [24]. That is to say, we use single type feature including GIST, HOG, LBP, CTM and Sifts-SPM as well as multiple type features fusion which 'mixed all' here. Among these, Sifts-SPM is based on local sifts feature and SPM [25] kernel function. In this experiment we use twenty-pass strategy for single feature and twenty-five-pass for "Mix-all" cases considering that many more parameters are to be optimized for "Mix-all" than single feature. It takes more samples for obtaining more optimized results when the parameter vector is longer (multi-pass experimental strategy will be described in Section 4.1.2). Results are illustrated in Table 1. The AP in the table stands for mean value over all the classes.

Table 1
Comparison of batch and online-mode. 'Mix-all' stands for 'GIST+HOG+LBP+CTM +Sifts-SPM'.

Feature	Ap. on dataset A		Ap. on dataset B	
	BCGD (%)[18]	OLHFFM (%)	BCGD (%)[18]	OLHFFM (%)
GIST	64.95	**70.14**	66.06	**69.76**
HOG	48.34	**50.18**	54.08	**55.94**
LBP	**62.53**	60.54	**70.40**	61.06
CTM	**65.80**	65.01	**76.42**	73.18
Sifts-SPM	77.54	**80.33**	**86.06**	81.98
Mix-all	**84.14**	82.04	**88.00**	85.64

As shown in Table 1, OLHFFM outperforms batch BCGD algorithm in cases of single feature including GIST and HOG on both datasets. However, when using LBP, CTM on both datasets, OLHFFM is not as good as BCGD in terms of classification performance. When using Sifts-SPM, OLHFFM results in more than two percentage-points higher AP than BCGD on dataset A but it suffers 4.08 percentage point decrease on dataset B. This suggests that for OLHFFM, the sensitivity to the same feature is not the same for different datasets. In addition, when we 'mix all', OLHFFM is slightly inferior to BCGD. There is 2.10% and 2.36% point decrease in AP value when comparing OLHFFM with BCGD on dataset A and B, respectively. To summarize, with the adoption of feasible multiple-pass strategy, the performance of OLHFFM is very close to that with batch solution BCGD. Due to the robustness gained by using multiple passes, the algorithm could be extended in online setting without any significant loss in performance.

4.1.2. Effect of multi-pass strategy

Due to relatively slow convergence rate of online algorithm, it is difficult to obtain good hypothesis after the training examples are used up. This is especially true when the number of training instances is small. So we try to enlarge the training dataset by cycling through the training examples several passes. Different passes correspond to different permutations of the training set.

How does multi-pass strategy affect recognition performance? We will study the effect in this experiment. The experimental results are illustrated in Fig. 1. The overall trend is that AP value increases with the increase number of passes for single feature or combinations of multi-features. This upward trend is more obvious at the beginning of the curve. With 'PassNum' (the number of passes) becomes bigger, AP value becomes more stabilized. This suggests that using OLHFFM better generalization performance is obtained after several passes and it reaches a reasonable level of convergence with increased number of passes. From Fig. 1, one can see the fluctuations in the trend lines, which is probably associated with the characteristics of online learning algorithm. In most cases the improvement reaches a plateau after 20 passes.

4.2. Evaluations on object categorization

Two benchmark object recognition datasets including MIT Caltech 6-categories and VOC2011 PASCAL TrainVal dataset are used for evaluating object categorization. The MIT Caltech 6-categories dataset contains 5775 images [41,42]. The image

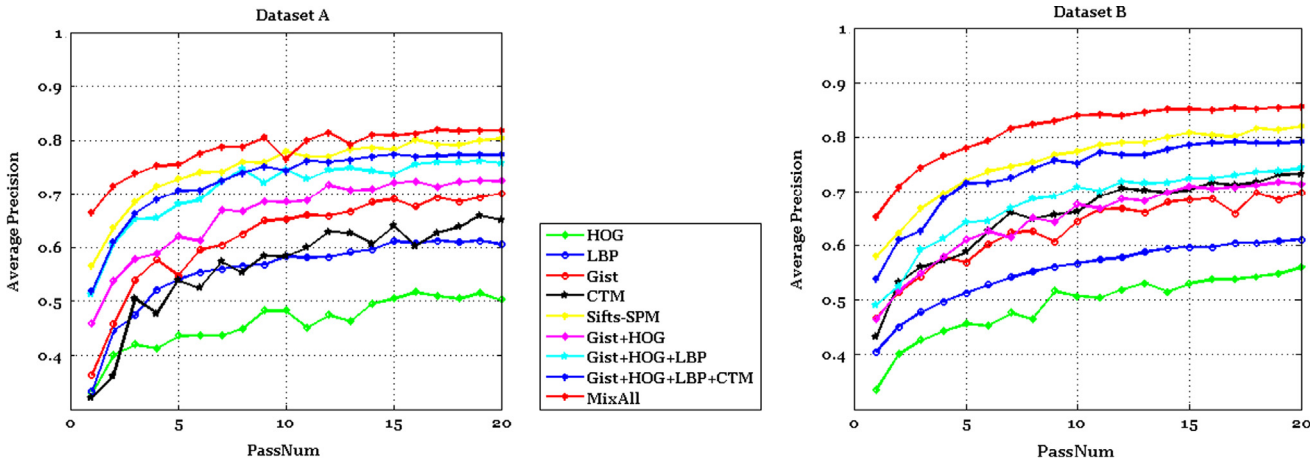

Fig. 1. Effectiveness of the number of refining pass.

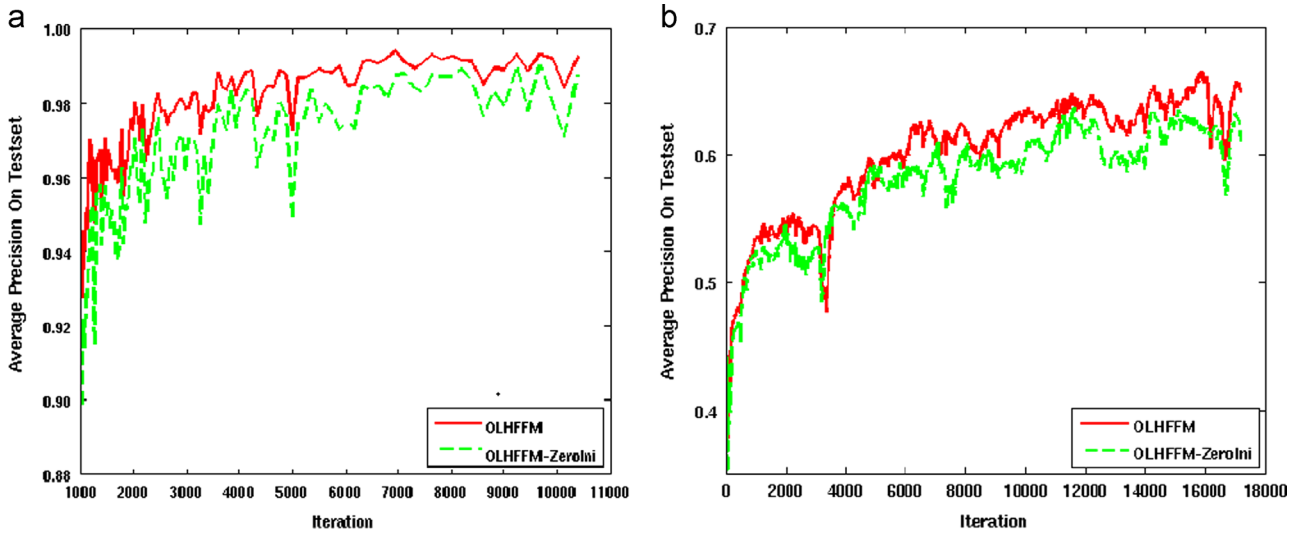

Fig. 2. Comparison between OLHFFM and OLHFFM-ZeroIni. (a) Learning curves on 6-categories Caltech and (b) learning curves on VOC2010 TrainVal.

number for each category is: 1074 aero-planes, 450 faces (front views of faces), 1155 cars (Side), 826 motorbikes, 1370 car backgrounds and 900 general background scenes. We select randomly 60% from each class for training and the remainder for test. The VOC2011 PASCAL TrainVal dataset [43] includes 11,540 images among which 5717 are for training and the remaining 5823 are for validation. Here we use the validation set for testing. Either training or validation set covers total 20 classes including aeroplane, bicycle and etc. It is a challenging database because there are a large number of samples with multi-label, occlusion, scale variant and cluster. Here we take category "aeroplane" as experimental subject. From the web [51], 327 of 5717 images are 'aeroplane' and the rest covers all the other 19 classes; 343 of 5823 images are 'aeroplane' and the rest belongs to other 19 classes. In the experiment, the whole VOC2011 PASCAL TrainVal dataset acted as experimental subject. That is to say, all the 670 "aeroplane" images are used as positive samples and the rest are used as negative ones.

In the experiment that is conducted on VOC2011 PASCAL TrainVal we use 'Mix-all' as described in Section 4.1. That is to say, we mix all the five types of features including Sifts-SPM, GIST, LBP, CTM, and HOG. However, very high AP value close to 1 can easily be achieved on the 6-categories Caltech dataset especially when we use more of the features e.g. GIST, Sifts-SPM etc. In order to illustrate the difference and make us see clearly, only "HOG +CTM" combination is adopted for the 6-categories Caltech dataset. We train the algorithms by cycling up to 3 passes over the training set to obtain the data size of over 10,000.

4.2.1. *Effect of value-added rule*

As stated in Section 3, we design reasonable added value principle for calculating the running average sub-gradient. The goal is to see if OLHFFM would incorporate dynamically the knowledge of the observed data in earlier iterations to perform more informative gradient-based online learning. Here we try to explore how the value-added approach affects the classification performance of OLHFFM.

In the following experiment, we try not to make the initial value of newly added row of gradient matrix g , but to pad it with zero for the consistency of the gradient matrix structure. The algorithm in this situation is called OLHFFM-ZeroIni. We compare OLHFFM and OLHFFM-ZeroIni on both 6-categories Caltech and VOC2011 PASCAL TrainVal datasets. The results of verification experiment are shown in Fig. 2 which shows AP value as a function of the number of iteration.

From Fig. 2, we see that the trend curve for OLHFFM and OLHFFM-ZeroIni are basically the same. But OLHFFM curve is always above that of OLHFFM-ZeroIni. That demonstrates that our value-added principle is a reasonable approach for taking into

consideration the impact on the current model update from the past samples.

4.2.2. Comparison with ILK and NORMA

We compare OLHFFM with two single kernel-based online algorithms, NORMA [25] and ILK [26] after they are extended to multi-kernel situation using averaging kernel strategy. Among these three algorithms, NORMA is a standard kernel-based stochastic sub-gradient method that largely follows a predetermined procedural scheme. It performs gradient with respect to the instantaneous risk at a constant learning rate. ILK and OLHFFM, on the other hand, solve simple optimization problems that integrate knowledge about Bregman divergence, loss function and regularization item. So they update model parameters at each round based on analytical solutions to simple constrained optimization problem. This unified view makes it meaningful to compare ILK with OLHFFM. In all of these three algorithms, online learning and kernel parameters are determined via cross-validation for optimized solution.

Fig. 3 shows AP value as a function of iteration number. Increase trend is observed on all the 'iteration-AP' curves of three algorithms. This indicates that the best solution is constantly tracked while using all these three algorithms. From Fig. 3, two obvious advantages of using OLHFFM can be found. First, its "iteration-AP" curve is flatter than the other two. This suggests that it provides more stability as the iteration goes. This effect is especially obvious as shown in Fig. 3 (a). It suggests that more stable solution can be achieved by using OLHFFM approach. Second, "iteration-AP" curve of OLHFFM algorithm is above that of ILK and NORMA during most of the run time, which shows better performance of the OLHFFM after the initial oscillations have died out.

4.3. Evaluation on handwritten digit recognition

We also evaluated OLHFFM on handwritten digit recognition dataset MNIST. The MNIST database is available from webpage [44]. It has a training set of 60,000 examples and a test set of 10,000 examples. The 60,000 pattern training set contains examples from approximately 250 writers. The digits have been size-normalized and centered in a fixed-size gray-scale 28×28 pixel digit images. They differ drastically from those samples of such visual datasets as 6-categories Caltech, VOC2011. For example, digit images are gray-scale ones so no color cues are needed for the feature description. Sifts-SPM is based on dense grid sift and bag of words representation which is especially suitable for the case of complicated images with noisy background and dominant intra-class variety. However, these tiny digit images are pure and clean ones without noisy and complicated background. In addition, SPHOG feature is similar to PHOG and used widely in handwritten digit recognition, so we only select one of them as the feature description. To summarize, SPHOG, LBP and GIST are used to describe handwritten digits in the following experiments.

4.3.1. Classification performance via sparsity degree parameter

In this section, we show how the behavior of the OLHFFM algorithms changes when the sparsity controlling parameter λ changes. Table 2 lists the tradeoffs between sparsity and Average Precision value over all the classes.

From Table 2, several conclusions can be drawn. First, the classification performance evaluated by AP index is affected directly by sparsity parameter. In particular, the AP index goes as high as 0.9907 from 0.9536 when λ changes from 0.001 to 0.0001. Almost four percentage point higher is achieved in AP value which is not negligible for user experience in handwritten digit recognition. The AP value reaches the highest point when $\lambda = 1e{-}6$ and at this time the sparsity is 0.40%. This indicates that approximately 240 group coefficients have been set to zero by means of group LASSO. In other words, 240 samples are considered as noisy. Fig. 4 shows examples of some noisy samples. Second, AP value is not really sensitive when λ changes from $5e{-}4$ to$1e{-}7$. However, the sparsity rate decreases from 59.71% to 0% which means the complexity of the model increases dramatically. Since λ does not obviously influence AP value which measures recognition accuracy, we should set the appropriate value of λ to achieve tradeoffs between sparsity and Average Precision value. Third, the AP value does not reach the highest when the sparsity becomes zero. It means that there are indeed some noisy samples in the training set that could degrade recognition performance. To summarize,

Table 2
Sparsity and AP via λ.

λ	Sparsity (%)	AP
0.001	98.44	0.9536
0.0001	75.68	0.9907
0.00005	59.71	0.9931
0.00001	12.11	0.9934
0.000001	0.40	**0.9935**
0.0000001	0	0.9933

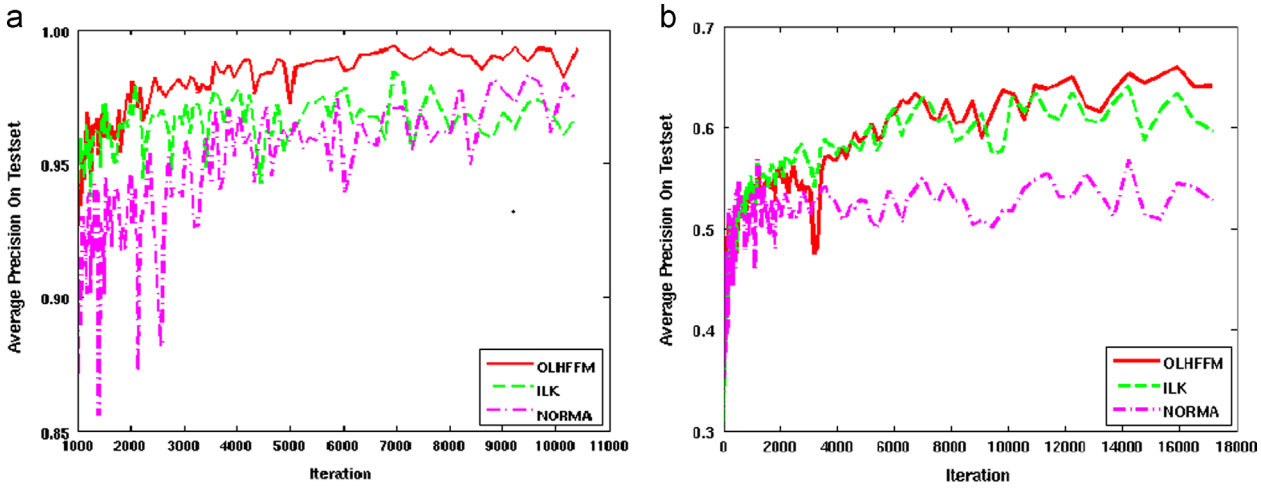

Fig. 3. Performance comparison of OLHFFM, ILK, NORMA. (a) Learning curves on 6-categories Caltech and (b) learning curves on VOC2010 TrainVal

appropriate parameter values have to be determined to balance between sparsity and classification performance.

4.3.2. Comparison with state-of-the-art multiple kernel learning algorithms

OLHFFM is compared against the state-of-the-art online multiple kernel learning algorithms such as UFO-MKL, OM-2, OMCL and OMKL. DOGMA package [45] is used for implementing OM-2, UFO-MKL, OMCL algorithms.

In Fig. 5, UFO-MKL-logistic and UFO-MKL-hinge are shown as two versions of UFO-MKL based on different loss functions. The first one is based on logistic loss and the second on hinge loss. Notations begin with 'OMKL' stands for 6 algorithms (from Algorithm 1 to Algorithm 6) in sequence as reference to Rong Jin's paper [15]. They are all under the same hierarchical online multiple kernel learning framework, i.e. OMKL. However, they are setup differently: one combines binary predictions denoted as 'P' and the other combines real value outputs denoted as 'O'. In both of the setup conditions, deterministic and stochastic approaches denoted as 'DA' and 'SUA or MUA' are used. It needs to be pointed out that OM-2 is proposed in particular for multi-class classification problems. So it is converted to binary classifier using binary hinge loss function in this experiment.

Fig. 4. Sparsified examples.

From Fig. 5, one can see that as far as average precision is concerned, OLHFFM ranks the fourth among the eleven algorithms. AP value averaged over ten digit class as high as 99.35% is reached with OLHFFM approach. It is very close to that from OM-2, UFO-MKL-logistic and OMCL. In fact, what is critical for OLHFFM algorithm is how to select important explanatory samples and eliminate those with less discriminant power by integrating group LASSO technique in kernel logistic regression model. On the contrary, OM-2 uses group p-norm to obtain a simpler optimization problem, however, the true sparsity is lost. That means that the weights of the kernels, even if they can become extremely small, will never be exactly zero. UFO-MKL-logistic presents elastic-net kind of regularization which mixes group p-norm and LASSO. It not only has the effect of inducing exact sparsity in the domain of the kernels but also makes the optimization problem easier. Both OM-2 and UFO-MKL-logistic define group based the coefficients related to the same kernel. That means both solutions will lead to the selection a subset of useful kernels. These kernel coefficients group based approach works well for problems with more features than the number of samples, where the common premise held there is that many features are irrelevant. On the other hand, such algorithm in which group is formed with the coefficients related to the same sample, like OLHFFM, is good for the case we often have too many samples [24]. In addition, OM-2, UFO-MKL and OMCL are all online solution for standard MKL problem which combine different kernels by a weighted summation to fusing multiple features. However, OLHFFM provides data-dependent weights to balance the contribution of each feature in a nonlinear fashion. It would improve fusion capability and describe possible nonlinear relationships among different types of features [24]. Finally, one point that makes OLHFFM superior is that it can be adapted feasibly to online solution of LASSO regularized logistic regression model based on single kernel.

5. Conclusion and future work

In this paper, a novel and efficient algorithm OLHFFM is presented. It can be used to solve heterogeneous feature fusion model based on multiple kernels in online setting. We conducted

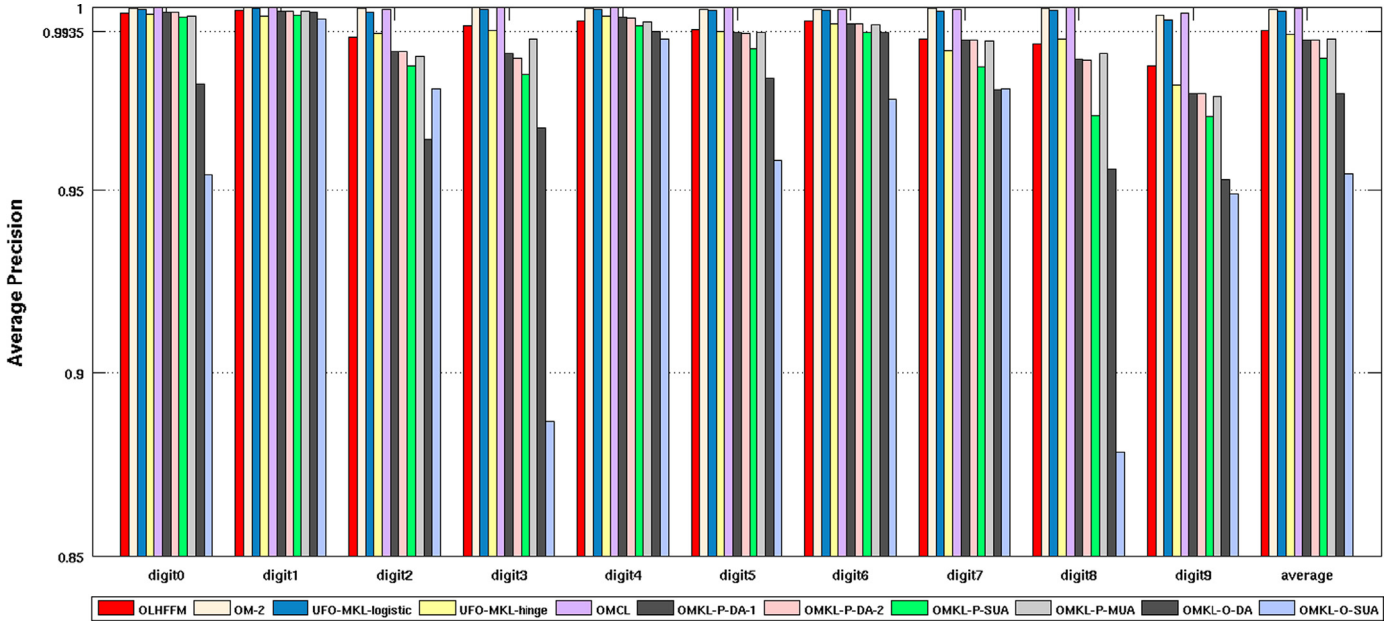

Fig. 5. Comparison with UFO-MKL, OM-2, OMCL, OMKL and their variants.

extensive experiments on a variety of visual recognition tasks that include event, object and handwritten digit recognition. Based on the experimental results we reached the conclusions that OLHFFM overcomes the inefficiencies of batch solution and it can be used to solve large-scale problems. Using OLHFFM approach, comparable accuracy as batch mode algorithms has been reached. In recognition performance, OLHFFM is as competitive as state-of-the-art approaches such as ILK, NORMA with averaged kernel and multiple kernel learning algorithms OM-2, UFO-MKL, OMCL and OMKL. In addition, group LASSO sparsity can be achieved even in the online setting with kernels to reduce model complexity. As coefficients related with the same sample yet with different feature and kernel function are grouped, noisy samples will be removed from classifier model by setting the corresponding coefficient groups zero. These enhanced features of OLHFFM make it a viable alternative to the batch algorithms BCGD in large scale dataset.

Future work related to OLHFFM can be conducted in the following areas: (1) to apply it on very large scale dataset such as ImageNet [46]; (2) to limit the amount of memory required to store the online hypothesis which may increase without bound as the algorithm progresses by truncation or projection method; (3) to extend the algorithm so that it can deal with multi-class problem; (4) to solve other kernel-based classification or regression model such as square loss, hinge loss and so on; (5) to make it suitable to other composite regularization method like sparse groupLASSO and explore the effectiveness for visual recognition; and (6) to theoretically analyze convergence rates and error bounds for this OLHFFM algorithm.

Acknowledgment

This research is supported in part by the National Natural Science Foundation of China (Grant no.: 61075021), the National Science and Technology Support Plan (2013BAH65F01-2013BAH65F04), the Guangdong Natural Science Funds (Grant no. S2011020000541), the GDSTP (No. 2012A010701001), and the Fundamental Research Funds for the Central Universities of China (Nos.2012ZP0002 and D2116320).

References

[1] Dacheng Tao, Xiaoou Tang, Xuelong Li, Xindong Wu, Asymmetric Bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1088–1099.
[2] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of International Conference on Computer Vision, (1999) pp. 1150–1157.
[3] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vision 42 (3) (2001) 145–175.
[4] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, vol. 2 (2005) pp. 886–893.
[5] T. Ojala, M. Pietikainen, T Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[6] H. Yu, M. Li, H. Zhang. J. Feng, Color texture moments for content-based image retrieval, in: Proceedings of International Conference on Image Processing, vol. 3 (2002) pp. 929–932.
[7] S. Belongie, J. Malik, J. Puzicha, Shape context: a new descriptor for shape matching and object recognition, in: Proceedings of Advances in Neural Information Processing Systems, (2000).
[8] Xuelong Dacheng Tao, Xindong Li, Stephen J. Wu, Maybank, general tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.
[9] Yun Fu, Liangliang Cao, Guodong Guo, Thomas S. Huang, Multiple Feature fusion by subspace learning, in: Proceedings of the International conference on Content-based Image and Video Retrieval, (2008) pp. 127–134.
[10] Ning Chen, Jun Zhu, Eric P. Xing, Predictive subspace learning for multi-view data: a large margin approach, Proceedings of Advances in Neural Information Processing Systems, (2010) 361–369.
[11] J. Luo, Francesco Orabona, Marco Fornoni, Barbara Caputo, Nicolò Cesa-bianchi, OM-2: an online multi-class multi-kernel learning algorithm, in: Proceedings of the 4th IEEE Online Learning for Computer Vision Workshop, (2010) pp. 43–50.
[12] R.Bach Francis, Consistency of the group LASSO and multiple kernel learning, J. Mach. Learn. Res. 9 (6) (2008) 1179–1225.
[13] Gilles Xilan Tian, Stephane Canu Gasso, A multiple kernel framework for inductive semi-supervised SVM learning, Neurocomputing 90 (1) (2012) 46–58.
[14] Francesco Orabona, Luo Jie, Ultra-fast optimization algorithm for sparse multi kernel learning, in: Proceedings of the 28 th International Conference on Machine Learning, (2011) pp. 249–256.
[15] Rong Jin, Steven C.H. Hoi, Tianbao Yang, Online multiple kernel learning: algorithms and mistake bounds, in: Proceedings of International Conference of the Association for Learning Technology, (2010) pp. 390–404.
[16] Tian Xia, Dacheng Tao, Tao Mei, Yongdong Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. B: Cybern. 40 (6) (2010) 1438–1446.
[17] B. Long, P. S. Yu, Z. Zhang, A general model for multiple view unsupervised learning, in: Proceedings of the SIAM International Conference on Data Mining, (2008) pp. 822–833.
[18] A. Argyriou, M. Herbster, M. Pontil, Combining graph laplacians for semi-supervised learning, in: Proceedings of Advances in Neural Information Processing Systems, (2005).
[19] K. Tsuda, H. Shin, B Scholkopf, Fast protein classification with multiple networks, Bioinformatics 21 (2) (2005) 59–65.
[20] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: Proceedings of the Workshop on Learning with Multiple Views, 22nd ICML, (2005).
[21] Yang Bo Xie, Dacheng Mu, Kaiqi Huang Tao, m-SNE: multiview stochastic neighbor embedding, IEEE Trans. Syst. Man Cybern. B: Cybern. 41 (4) (2011).
[22] Xuelong Dacheng Tao, Xindong Li, Stephen J.Maybank Wu, Geometric Mean for Subspace Selection, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 260–274.
[23] Dongpo Hongmei Shao, Gaofeng Xu, Lijun Liu Zheng, Convergence of an online gradient method with inner-product penalty and adaptive momentum, Neurocomputing 77 (1) (2012) 243–252.
[24] Liangliang Cao, Jiebo Luo, Feng Liang, Thomas S. Huang, Heterogeneous Feature Machines for Visual Recognition, in: Proceedings of International Conference on Computer Vision, (2009) pp. 1095–1102.
[25] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (2006).
[26] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (1958) 386–407.
[27] J. Duchi, Y. Singer, Efficient Learning using Forward-Backward Splitting, in: Proceedings of Advances in Neural Information Processing Systems, (2009) pp. 495–503.
[28] L. Xiao, Dual Averaging method for regularized stochastic learning and online optimization, in: Proceedings of Advances in Neural Information Processing Systems, (2009) pp. 2116–2124.
[29] Haiqin Yang, Zenglin Xu, Irwin King, Michael R. Lyu, Online learning for group lasso, in: Proceedings of International Conference on Machine Learning, (2010) pp. 1191–1198.
[30] J. Kivinen, A.J. Smola, R.C. Williamson, Online learning with kernels, IEEE Trans. Signal Process. 100 (10) (2010) 1–12.
[31] S.V.N. Li Cheng, Vishwanathan, Dale Schuurmans, Shaojun Wang, Terry Caelli, Implicit online learning with kernels, in: Proceedings of Advances in Neural Information Processing Systems, (2007) pp. 249–256.
[32] Luo Jie, Francesco Orabona, Barbara Caputo, An online framework for learning novel concepts over multiple cues, in: Proceedings of Asian Conference on Computer Vision, (2009).
[33] Francesco Orabona, Luo Jie, Barbara Caputo, Online-batch strongly convex mUlti kernel learning, in: Proceedings of IEEE Conference of Computer Vision and Pattern Recognition, (2010).
[34] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B 68 (1) (2006) 49–67.
[35] P. Tseng, S. Yun, A coordinate gradient descent method for nonsmooth separable minimization, Math. Programming B 117 (1–2) (2009).
[36] L.M. Bregman, The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming, USSR Comput. Math. Math. Phys. 7 (1967) 200–217.
[37] B. Scholkopf, R..Herbrich, A.J. Smola, A generalized representer theorem, in: Proceedings of the Annual Conference on Computational Learning Theory, (2001) pp. 416–426.
[38] L.-J. Li, L. Fei-Fei, What, where and who? classifying event by scene and object recognition, in: Proceedings of IEEE International Conference on Computer Vision, (2007).
[39] V. Jain, A. Singhal, J. Luo, Selective hidden random fields: exploiting domain specific saliency for event classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (1) (2008).
[40] J. Weston, A. Bordes, L. Bottou, Online (and offline) on an even tighter budget, in: Proceedings of AISTATS, (2005) pp. 413–420.
[41] ⟨http://www.robots.ox.ac.uk/∼vgg/data/⟩.
[42] ⟨http://l2r.cs.uiuc.edu/∼cogcomp/index⟩ research.html.
[43] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results, (2011).

[44] ⟨http://yann.lecun.com/exdb/mnist/⟩.
[45] Francesco Orabona, DOGMA: a MATLAB toolbox for Online Learning, (2009).
[46] J. Deng, W. Dong, R. Socher, L. Jia Li, K. Li, L.Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (2009) pp. 248–255.
[47] Cao Liangliang, Heterogeneous Feature Fusion for Visual Recognition, University of Illinois at Urbana-Champaign, 2011, PhD thesis.
[48] S. Kakade, S. Shalev-Shwartz, A. Tewari, On the duality of strong convexity and strong smoothness: learning applications and matrix regularization. Technical Report, TTI, 2009.
[49] S. Shalev-Shwartz., Online Learning: Theory, Algorithms, and Applications, The Hebrew University,, 2007, PhD thesis.
[50] Xiao Lin, Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization, J. Mach. Learn. Res. 11 (2010) 2543–2596.
[51] ⟨http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/dbstats.html⟩.

Shuangping Huang received M.S. degree and Ph.D. degree from the South China University of Technology in 2005 and 2011, respectively. She is currently a lecturer in the College of Engineering at South China Agricultural University, Guangzhou, China. Her research interests include machine learning, computer vision and data mining.

Lianwen Jin received a BS degree from the University of Science and Technology of China and a Ph.D. degree from South China University of Technology in 1991 and 1996, respectively. He is now a professor with the School of Electronic and Information Engineering, South China University of Technology. He is the author of more than 100 scientific papers. He is member of IEEE Signal Processing Society, IEEE Communication Society, and IEEE Computer Society, China Image and Graphics Society, the Cloud Computing Experts Committee of China Institute of Communications. He received the award of New Century Excellent Talent Program of MOE in 2006, the Guangdong Pearl River Distinguished Professor award in 2011, respectively. He served as Program Committee member for a number of international conferences, including ICMLC2007-2011, ICFHR2008-2012, ICDAR2009, ICDAR2013, ICPR2010, ICPR2012, ICMLA2012 etc. His research interests include image processing, handwriting analysis and recognition, machine learning, cloud computing, and intelligent systems.

Yuan Fang received the B.S. degree in Electronics and Information Engineering from South China Agricultural University, Guangzhou, China. She is currently a Master candidate in Computer Science at Northeastern University, Boston, U.S. Her research interests include machine learning and computer vision.

Xiaoxin Wei received B.S. degree from South China Agricultural University in 2011. She is currently a M.S. candidate in the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. Her research interests include character recognition, machine learning and computer vision.