Neurocomputing 185 (2016) 171–182

Contents lists available at ScienceDirect

Neurocomputing

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

A fast alternating time-splitting approach for learning partial differential equations

Zhenyu Zhao [a], Zhouchen Lin [b,c,*], Yi Wu [a]

[a] College of Science, National University of Defense Technology, PR China
[b] Key Lab. of Machine Perception (MOE), School of EECS, Peking University, PR China
[c] Cooperative Medianet Innovation Center, Shanghai Jiaotong University, PR China

ARTICLE INFO

Article history:
Received 3 October 2014
Received in revised form
13 May 2015
Accepted 20 October 2015
Communicated by T. Heskes
Available online 30 December 2015

Keywords:
Learning-based PDEs
PDE constrained optimal control
FATSA
Computer vision
Image processing

ABSTRACT

Learning-based partial differential equations (PDEs), which combine fundamental differential invariants into a nonlinear regressor, have been successfully applied to several computer vision and image processing problems. However, the gradient descent method (GDM) for solving the linear combination coefficients among differential invariants is time-consuming. Moreover, when the regularization or constraints on the coefficients become more complex, it is troublesome or even impossible to deduce the gradients. In this paper, we propose a new algorithm, called fast alternating time-splitting approach (FATSA), to solve the linear combination coefficients. By minimizing the difference between the expected output and the actual output of PDEs *at each time step*, FATSA can solve the linear combination coefficients much faster than GDM. More complex regularization or constraints can also be easily incorporated. Extensive experiments demonstrate that our proposed FATSA outperform GDM in both speed and quality.

1. Introduction

Partial differential equations (PDEs) have been successfully used to solve many practical problems in computer vision and image processing [1–3], such as denoising [4,5], enhancement [6,7], inpainting [8], segmentation [9,10], and optical flow computation [11,12]. However, it is usually difficult to design a PDE system for a particular task which requires high mathematical skills and good insight into the problem. According to [13], the existing methods of designing PDEs can be mainly classified into two groups. The methods of the first group write down PDEs directly, which requires good mathematical understandings on the properties of the PDEs. The methods of second group first define an energy functional [14], which pursues the expected properties of the output image or video, and then derive evolution equations by computing the Euler–Lagrange variation of the energy functional. For example, the ROF model [15] and TV-L_1 [16] for image denoising are designed directly, while the Nambu model [17] and the PL model [18] for color image processing are designed in the variational way.

To reduce the difficulty in designing PDEs for complex vision problems, Liu and Lin et al. [13] proposed a framework that learns PDEs from training image pairs recently. They first considered learning PDEs for grayscale image restoration [19], which involve an anisotropic diffusion term. Then they generalized the idea significantly by linearly combining fundamental differential invariants that are invariant to translation and rotation. These differential invariants serve as “bases” of differential operators [13,20]. They utilized the gradient descent method (GDM) to solve the linear combination coefficients. The learnt PDEs have been successfully applied to various problems, such as image denoising, debluring, object detection, color to gray and demosaicking [13,21,22].

However, GDM has several drawbacks. First, the convergence speed of GDM is very slow due to the fact that objective functional is flat. Experiments show that the magnitude of gradient is usually at the order of 10^{-3} (Fig. 4), even at the beginning iterations. Therefore, the solution of GDM does not improve the initial value very much. Second, it needs to solve the adjoint PDEs to obtain the gradient, which is difficult to deduce and also time-consuming. Third, when the regularization or constraints on the linear combination coefficients become more complex, e.g., we use L_1 norm as the regularizer or add boundedness constraints, the deduction of gradient becomes very involved or even non-existent because of the non-differentiablity of the objective functional. Last, the quality of learnt PDEs is not very good. For example, the magnitudes of coefficients are unbalanced. We can see from Fig. 4(a) that most of $a_i(t)$'s are close to zeros while some jump to more than 20. This can cause numerical instability as the differential invariants

* Corresponding author.
E-mail addresses: dwightzzy@gmail.com (Z. Zhao), zlin@pku.edu.cn (Z. Lin), wuyi_work@sina.com (Y. Wu).

http://dx.doi.org/10.1016/j.neucom.2015.10.126
0925-2312/

involve multiplications of second order derivatives. As a result, blowup might occur when we apply the learnt PDEs to test images. Moreover, we can also see from Fig. 4(c) that $b_i(t)$'s are very close to zeros, which means that the indicator function is actually ineffective.

To overcome the above short-comings of GDM, we propose a new method, called fast alternating time-splitting approach (FATSA), to solve the linear combination coefficients. We first discretize the PDEs in time. Then we minimize the difference between the expected output (ground truth) and the actual output of the PDEs at each time step n, which is a nonlinear regression problem and can be solved by alternately minimizing $a_i(n\Delta t)$'s and $b_j((n-1)\Delta t)$'s. In such a greedy manner, the linear combination coefficients can be updated sequentially in time. Based on FATSA, it is convenient to add constrains and regularization on the coefficients, even when these constrains and regularization are non-differentiable. Moreover, we do not need to deduce and compute the adjoint PDEs any longer. Besides, compared with GDM, FATSA can greatly reduce the training time and the training error. For grayscale images, the speed of training is accelerated by ten times. For color images, the training time is cut by half. In summary, the contributions of this paper are summarized as follows:

- We propose a new fast alternating time-splitting approach (FATSA) to solve the PDE constrained optimal control problem, which not only speeds up the learning process, but also improves the results.
- Compared with GDM, FATSA is much simpler. It computes the linear combination coefficients in temporal order. It avoids to compute the adjoint PDEs for evaluating the Gâteaux derivatives [23] of the objective functional.
- FATSA is much more flexible than GDM. When we add more general regularizations (e.g., non-smooth regularization) and extra constraints on the linear combination coefficients, it can also improve the results.

The rest of the paper is organized as follows. First of all, we review the learning-based PDEs methods briefly in Section 2. Then we present the main idea of FATSA and the details of alternating minimization in Section 3. In Section 4, we discuss the complexity of FATSA and make a detailed comparison with GDM [13]. We also extend FATSA to solve learning-based PDEs for vector-valued image processing problems. Then in Section 5, we compare the performance of FATSA and GDM on some computer vision and image processing problems. Finally, we conclude our paper and discuss the future work in Section 6.

2. Learning-based PDEs

In this section, we briefly review the framework of learning-based PDEs for computer vision and image processing problems. More details can be found in [13,21,22].

2.1. Mathematical formulation

The current learning-based PDEs are grounded on the translational and rotational invariance of computer vision and image processing problems. Namely, when the input image is translated or rotated, the output image should be translated or rotated accordingly. Then it can be proven that the governing equations are functions of fundamental differential invariants, which form “bases” of all differential invariants that are invariant with respect to translation and rotation. We assume that the evolution of the image u is guided by an indicator function v, which collects large scale information. As shown in Table 1, there are 17 fundamental differential invariants $\{\mathrm{inv}_i(u,v),\ i=0,\dots,16\}$ up to the second order. For brevity, we denote $\mathbf{inv}(u,v)=[\mathrm{inv}_0(u,v),\mathrm{inv}_1(u,v),\dots,\mathrm{inv}_{16}(u,v)]^T$, where $(\cdot)^T$ denoted the transpose of matrix (or vector).

Table 1
Fundamental differential invariants up to the second order, where tr is the trace operator and ∇f and $\mathbf{H}_f$ are the gradient and the Hessian matrix of function f, respectively.

j	$\mathrm{inv}_j(u,v)$
0,1,2	$1, v, u$
3,4	$\|\nabla v\|^2=v_x^2+v_y^2,\ \|\nabla u\|^2=u_x^2+u_y^2$
5	$(\nabla v)^T\cdot\nabla u=v_xu_x+v_yu_y$
6,7	$\mathrm{tr}(\mathbf{H}_v)=v_{xx}+v_{yy},\ tr(\mathbf{H}_u)=u_{xx}+u_{yy}$
8	$(\nabla v)^T\cdot\mathbf{H}_v\cdot\nabla v=v_x^2v_{xx}+2v_xv_yv_{xy}+v_y^2v_{yy}$
9	$(\nabla v)^T\cdot\mathbf{H}_u\cdot\nabla v=v_x^2u_{xx}+2v_xv_yu_{xy}+v_y^2u_{yy}$
10	$(\nabla v)^T\cdot\mathbf{H}_v\cdot\nabla u=v_xu_xv_{xx}+(v_xu_y+u_xv_y)v_{xy}+v_yu_yv_{yy}$
11	$(\nabla v)^T\cdot\mathbf{H}_u\cdot\nabla u=v_xu_xu_{xx}+(v_xu_y+u_xv_y)u_{xy}+v_yu_yu_{yy}$
12	$(\nabla u)^T\cdot\mathbf{H}_v\cdot\nabla u=u_x^2v_{xx}+2u_xu_yv_{xy}+u_y^2v_{yy}$
13	$(\nabla u)^T\cdot\mathbf{H}_u\cdot\nabla u=u_x^2u_{xx}+2u_xu_yu_{xy}+u_y^2u_{yy}$
14	$\mathrm{tr}(\mathbf{H}_v^2)=v_{xx}^2+2v_{xy}^2+v_{yy}^2$
15	$\mathrm{tr}(\mathbf{H}_v\cdot\mathbf{H}_u)=v_{xx}u_{xx}+2v_{xy}u_{xy}+v_{yy}u_{yy}$
16	$\mathrm{tr}(\mathbf{H}_u^2)=u_{xx}^2+2u_{xy}^2+u_{yy}^2$

The simplest function of fundamental differential invariants is a linear combination of them. Therefore, learning the PDEs can be transformed into learning the linear combination coefficients among the fundamental differential invariants, which are functions of time t only and independent of spatial variables [13,21,22]. To this end, one may prepare a number of input/output training image pairs. By minimizing the difference between the output of PDEs and the ground truth. We set the initial function as the input image. This results in a PDEs constrained optimal control problem:

$$\min_{\mathbf{a},\mathbf{b}} E(\mathbf{a}(t),\mathbf{b}(t))=\frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}(O_m-u_m(x,y,T))^2\,\mathrm{d}\Omega + \lambda_1\sum_{i=0}^{16}\int_0^T a_i^2(t)\,\mathrm{d}t+\lambda_2\sum_{i=0}^{16}\int_0^T b_i^2(t)\,\mathrm{d}t, \tag{1}$$

$$\text{s.t.}\quad \begin{cases} \dfrac{\partial u_m}{\partial t}-\mathbf{inv}^T(u_m,v_m)\cdot\mathbf{a}(t)=0, & (x,y,t)\in Q,\\ u_m(x,y,t)=0, & (x,y,t)\in\Gamma,\\ u_m(x,y,0)=I_m, & (x,y)\in\Omega,\\ \dfrac{\partial v_m}{\partial t}-\mathbf{inv}^T(v_m,u_m)\cdot\mathbf{b}(t)=0, & (x,y,t)\in Q,\\ v_m(x,y,t)=0, & (x,y,t)\in\Gamma,\\ v_m(x,y,0)=I_m, & (x,y)\in\Omega, \end{cases} \tag{2}$$

where $\{(I_m,O_m),\ m=1,\dots,M\}$ denote the M input/output training image pairs, $u_m(x,y,t)$ is the evolution image at time t with respect to the input image I_m, $v_m(x,y,t)$ is the corresponding indicator function, $\Omega\subset\mathbb{R}^2$ is the (rectangular) region occupied by the image,[1] T is the temporal span of evolution which can be normalized as 1, $Q=\Omega\times[0,T]$, $\Gamma=\partial\Omega\times[0,T]$, and $\partial\Omega$ denotes the boundary of Ω. The last two terms in (1) are regularization terms on the coefficients $a_i(t)$ and $b_i(t)$. We denote it as $\mathbf{a}(t)=[a_0(t),a_1(t),\dots,a_{16}(t)]^T$ and $\mathbf{b}(t)=[b_0(t),b_1(t),\dots,b_{16}(t)]^T$ for brevity. For $\mathrm{inv}_i(v,u)$, it can be acquired by simply switching u and v in $\mathrm{inv}_i(u,v)$.

[1] The images are padded with zeros of several pixels width around them, so that the Dirichlet boundary conditions, $u_m(x,y,t)=0, v_m(x,y,t)=0, (x,y,t)\in\Gamma$, are naturally fulfilled.