Self-Paced Dictionary Learning for Image Classification

Ye Tang, Yu-Bin Yang, Yang Gao
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing 210093, China
yangyubin@nju.edu.cn

ABSTRACT

Image classification is an important research task in multimedia content analysis and processing. Learning a compact dictionary easying to derive sparse representation is one of the focused issues in the state-of-the-art image classification framework. Most existing dictionary learning approaches assign equal importance to all training samples, which in fact have different complexity in terms of sparse representation. Meanwhile, the contextual information "hidden" in different samples is ignored as well. In this paper, we propose a self-paced dictionary learning algorithm in order to accommodate the "hidden" information of the samples into the learning procedure, which uses the easy samples to train the dictionary first, and then iteratively introduces more complex samples in the remaining training procedure until the entire training data are all easy samples. The algorithm adaptively chooses the easy samples in each iteration, while the learned dictionary in the previous iteration is in turn used as a basis for the current iteration. This strategy implicitly takes advantage of the contextual relationships among training samples. The number of the chosen samples in each iteration is determined by an adaptive threshold function proposed in this paper. Experimental results on benchmark datasets, including Caltech-101 and 15-Scene, show that our algorithm leads to better dictionary representation and classification performance than the baseline methods.

Categories and Subject Descriptors

I.4.8 [**Scene Analysis**]: Object recognition

General Terms

Algorithms, Performance, Experimentation

Keywords

Self-paced, dictionary learning, image classification

MM'12, October 29–November 2, 2012, Nara, Japan.

1. INTRODUCTION

In recent years, with the exponential growth of the variety of web images, efficient retrieval for the huge amount and diverse image data has become a grand challenging task. To tackle this issue, many researchers are now turning to study image classification [3, 6, 14, 17], which is one of the fundamental challenges in computer vision and targets at automatically assigning predefined class labels to images based on their visual features.

Dictionary learning, which has attracted more and more researchers' interests, is one of most successful method used in image classification tasks [16, 11]. It models input data with linear combinations of a few bases of a learned dictionary, and leads to sparse representation for input data. The performances of the state-of-the-art methods in the literatures [15, 7] have shown that sparse representation is very suitable for representing image data. Particularly, for a dataset $X \in R^{m\times n}$, this problem seeks for an optimal dictionary $D \in R^{m\times k}$ in which each item $x_i \in R^m$ in X can be well represented by a sparse linear combination of columns. Quite a few efficient algorithms aiming to find such a dictionary have been proposed in the past years. For instance, Yang et al. [15] proposed the ScSPM, a hierarchical model inspired by the *Bag-of-Features* model, which adopted the efficient sparse coding algorithm proposed in [10] to learn a compact dictionary on the local SIFT descriptors densely sampled from images. After obtaining the sparse coefficients of local descriptors, a spatial pyramid max pooling step was used to form the final image representation. Although the ScSPM achieves state-of-the-art performance, it is computational expensive to effectively solve the corresponding optimization problem, especially for large-scale datasets. Mairal et al. [12] presented an online dictionary learning algorithm aiming to minimize the expected cost instead of the empirical cost. Based on first-order stochastic gradient descent, the algorithm scaled up gracefully for large-scale dataset. In order to further improve the performance, supervised information such as class labels, has been used in recent work. A back-propagation based dictionary learning approach was presented in [16], which adopted linear classifier and attempted to minimize the classification error. Mairal et al. [11] proposed a general formulation and an efficient algorithm for supervised dictionary learning, which was adapted in a wide variety of tasks. In [7], a label-consistent K-SVD (LC-KSVD) method was presented to learn a discriminative dictionary. By introducing the constraint of "discriminative sparse-code error" and combining it with the reconstruction error and the classification error,

the algorithm assigned each feature with the same class labels by using similar sparse codes.

However, most existing methods treat dictionary learning as a one-pass "batch" procedure, by implicitly assuming that all training samples are equally important in learning, and all information relevant to learning is available at the first beginning. Furthermore, the contextual relationships among samples are ignored as well. Recent work in [2] presented curriculum learning for training with non-convex objectives and advocated a well-defined training order which used easy samples first and gradually expanded to more complex ones. But it required to identify the easy and the hard samples in advance, which is difficult in many real-world applications. In order to overcome this problem, Kumar et al. [8] proposed a self-paced learning mechanism for latent variable models. It simultaneously selected the easy samples and learned new parameters for a latent structural SVM. Inspired by the above work, we propose a self-paced dictionary learning algorithm, in which the self-paced learning mechanism is introduced into unsupervised dictionary learning procedure. To our best knowledge, this is for the first time that the contextual information among samples and the meaningful training order are considered simultaneously in the dictionary learning studies. Our approach iteratively chooses the easy samples as the training set for the current dictionary learning, in which any efficient dictionary learning algorithm can be employed. Furthermore, an adaptive threshold function is adopted to identify the easy samples, thus we may control the sampling procedure in each iteration flexibly. We also relax the sampling criterion as iteration increases, therefore more and more easy samples are selected until the whole training set is regarded as "easy". Our approach converges when all samples have been processed and the change of objective function is below the pre-defined tolerance value.

The rest of this paper is organized as follows. Section 2 presents the proposed algorithm in detail. Experimental results are reported in Section 3. Finally, we conclude this paper and discuss future work in Section 4.

2. SELF-PACED DICTIONARY LEARNING

2.1 Problem Statement

Given a finite set of the local descriptors extracted from training images $X = [x_1, ..., x_n] \in R^{m \times n}$, where $x_i \in R^m$ is the local descriptor, dictionary learning techniques used for image classification are intended to minimize the empirical cost function in Eq.(1).

$$f_n(D) \triangleq \frac{1}{n} \sum_{i=1}^{n} l(x_i, D), \tag{1}$$

where $D = [d_1, ..., d_k] \in R^{m \times k}$ is the dictionary, and $d_i \in R^m$ is a basis vector. Generally the dictionary is overcomplete, i.e. $k > m$. Suppose that $l(x, D)$ is a loss function which measures whether D is "good" at representing the local descriptor x. As shown in the ScSPM method [15], we consider $l(x, D)$ as the minimum of the l_1-sparse coding problem:

$$l(x, D) \triangleq \min_{\alpha \in R^k} ||x - D\alpha||_2^2 + \lambda ||\alpha||_1 \tag{2}$$

where l_1-norm yields a sparse solution for α, and λ is the regularization parameter balancing two items in Eq. (2). This sparse coding problem (also known as *Lasso* [13]) is extensively studied in the past few years, and many algorithms have been proposed to solve it, such as LARS [5].

Herewith, the dictionary learning problem can be rewritten as a joint optimization problem with respect to the dictionary D and the sparse codes $A = [\alpha_1, ..., \alpha_n] \in R^{k \times n}$.

$$\begin{aligned} &\min_{D,A} \tfrac{1}{n} \sum_{i=1}^{n} ||x_i - D\alpha_i||_2^2 + \lambda ||\alpha_i||_1 \\ &s.t. \quad ||d_j||_2^2 \leq 1, \quad j = 1, ..., k \end{aligned} \tag{3}$$

where the constraint is to prevent D from arbitrarily large. It is not jointly convex, but instead convex with respect to each of D and A when the other is fixed. Similar to most of the existing methods, we can alternately optimize between D and A by minimizing one while keeping the other fixed.

2.2 Self-Paced Dictionary Learning

Our algorithm employs a self-paced mechanism for dictionary learning. It enables the algorithm itself learn the dictionary using a fixed training order, i.e. from the easier samples to the harder ones. Throughout the learning process, we maintain two disjoint subsets of the original training set: (1) E, the easy samples chosen before the current iteration; and (2) H, the remaining hard samples before the current iteration. E and H are initialized as an empty set and the training set X respectively.

The proposed method iterates the following four main steps: (1) find the easy samples from H, and add them into E; (2) perform sparse coding by using E as the training set; (3) update the dictionary with fixed sparse coding results; (4) update the threshold used to identify the easy samples.

2.2.1 Finding the Easy Samples

In order to judge the "easiness" of a sample, we need to define a scoring function according to the current dictionary. A natural formulation is the objective function $l(x, D)$ shown in Eq.(2). Under this formulation, the easy samples correspond to small function values, while the hard samples relate to large function values. Consequently, we may choose an appropriate threshold σ to identify the easy samples. A sample x_i is viewed as "easy" if $l(x_i, D) \leq \sigma$. To avoid overfitting, we should set the initial threshold appropriately such that no less than half of the samples are considered "easy" in the first iteration. It is worth to note that other score function can also be used in our general framework, such as elastic-net formulation [18]:

$$l'(x, D) \triangleq \min_{\alpha \in R^k} ||x - D\alpha||_2^2 + \lambda_1 ||\alpha||_1 + \lambda_2 ||\alpha||_2^2 \tag{4}$$

2.2.2 Sparse Coding

The sparse coding stage searches the sparse coefficients α based on a signal x and a fixed dictionary D. The sparsest representation can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min_{\alpha} ||\alpha||_0 \\ &s.t. \quad ||x - D\alpha||_2^2 \leq \varepsilon \end{aligned} \tag{5}$$

where $\|\cdot\|_0$ denotes l_0 pseudo-norm, the number of nonzero elements of a vector. However, to accurately determine the sparsest representations proves to be an NP-hard problem [4]. A well known approximation approach is using convex l_1-norm to replace l_0 pseudo-norm as shown in Eq.(2).

Moreover, it has been empirically proven to be better in general sense. We use the efficient *feature-sign search* algorithm [10] to solve the l_1-norm sparse coding problem. The algorithm turns out to be adequately efficient to deal with large data sets.

2.2.3 Dictionary Update

With the fixed sparse coefficients A_E of the "easy" training samples, dictionary updating can be defined to solve a least square problem with the following quadratic constraints:

$$\begin{aligned} &\min_{D} ||X_E - DA_E||_2^2 \\ s.t. \quad &||d_j||_2^2 \ \leq 1 \quad j = 1, ..., k \end{aligned} \tag{6}$$

A Newton method on the Lagrange dual as used in [10] is adopted to update the dictionary.

2.2.4 Threshold Update

Unlike using a fixed annealing factor μ in [9] to update the threshold σ, i.e. $\sigma_{t+1} = \sigma_t \cdot \mu$, we regard μ as a monotone increasing function of iteration t, which guarantees more easy samples can be chosen as the iteration continues. We define the threshold as in the following equation:

$$\sigma = f(\pi, t) = \pi + \log(\pi^2 + c)t \quad (c \geq 1) \tag{7}$$

where π denotes the median of the objective function scores $l(x, D)$ of all training samples. This simple threshold function has at least three characteristics: (1) σ_0 satisfies the condition that no less than half of the samples are considered easy in the first iteration; (2) parameter c controls the growth rate of easy samples over time; (3) the threshold is more adaptive to the current dictionary and objective function value.

Our algorithm is detailed in **Algorithm 1**. Because $l(x, D)$ is used as the "easiness" of score function, our approach integrates the search of the easy samples and sparse coding into one step. On the whole, the algorithm draws the easy samples as training set, and solves two convex optimization problems in one iteration. As iteration continues, more and more easy samples are identified. When the entire training set are regarded as "easy", the algorithm becomes the standard dictionary learning algorithm for the training set, but it uses a well trained initial dictionary D_t. The algorithm will be terminated when the decrease of objective function in Eq.(1) is below a predefined tolerance ε.

As mentioned above, the optimization problem in Eq.(1) is non-convex with respect to D without fixing sparse coefficients. The alternating optimization strategy can decrease the value of empirical cost function monotonically, which makes the algorithm able to return a locally optimal dictionary. However, monotonicity seriously degrades the algorithm's convergence speed under some circumstances [1]. Instead, the self-paced dictionary learning algorithm uses only an easy subset for training in one iteration, and the value of the objective function no longer decreases all the time. Therefore, it is a kind of non-monotonic approach. Besides, the self-paced mechanism empirically leads to a better locally optimal solution to the non-convex problem in Eq.(1), which is consistent with the results achieved in [8].

Algorithm 1 Self-Paced Dictionary Learning

Input: Training samples $X = [x_1, ..., x_n] \in R^{m \times n}$; regularization parameter $\lambda \in R$; threshold parameter $c \in R$; initial dictionary $D_0 \in R^{m \times k}$
Output: Learned dictionary $D_{final} \in R^{m \times k}$

initialization: E =$\emptyset$; $H = X$; $t \leftarrow 0$
repeat
Find the easy samples and perform sparse coding using the dictionary D_t:
  for each $x_i \in X$ **do**
    perform sparse code using feature-sign search algorithm;
    if $l(x_i, D_t) < \sigma$
      x_i is an easy sample ;
      add it to E: $E = E \cup \{x_i\}$;
    end if
  end for
Update dictionary using Lagrange dual method;
Update the threshold using the function defined in Eq.(7);
$t \leftarrow t + 1$;
until All training samples have been processed as "easy" and the decrease of objective function in Eq.(1) is below tolerance ε.
return D_{final}

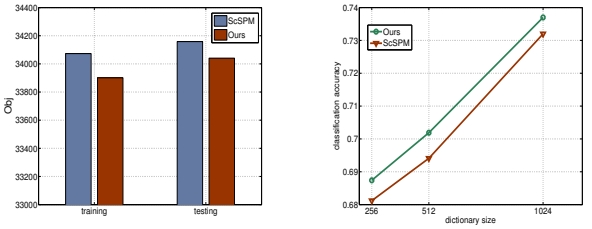
3. EXPERIMENTAL VALIDATION

In this section, we evaluate the proposed approach on the Caltech-101[1] and 15-Scene[2] dataset. More specifically, we implemented our algorithm and the ScSPM, in which the same sparse coding approach and dictionary updating strategy are adopted. The latter, together with Wang's algorithm are regarded as the baseline methods. The SIFT descriptors used in our implementation are extracted from 16×16 pixel patches and densely sampled on a grid with a step size of 8 pixels. We set the size of dictionary as 1,024 except the last experiment on Caltech-101. Furthermore, in order to select the easy examples at the first round of **Algorithm 1**, we need an appropriate initial dictionary. Accordingly, we compare two implementations of initial dictionary: one is obtained by K-means clustering and the other is constructed by performing 5 iterations of sparse coding and dictionary update on the entire data. Empirically, there are slight differences between them in terms of classification accuracy. Therefore, the initial dictionary obtained by K-means is used in our experiments. We set $\lambda = 0.15$ and $c = 1$ for all experiments. Our approach adopts max pooling to build the image representation and employs the linear SVM classifier for classification. The experiments are repeated 10 times with different random splits on the training and testing images in order to obtain reliable results, and the average classification accuracies are finally reported. By following the common experimental settings on the Caltech-101 dataset, we trained 5, 10, 15, 20, 25 and 30 samples per category and test the remainder. The detailed comparisons of the results are shown in Table 1. Our approach consistently outperforms the baseline methods, empirically demonstrating that the self-paced dictionary learning mechanism is able to lead to improved performance. Furthermore, in order to show

[1]http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[2]http://www.cs.unc.edu/lazebnik/research/scene_categories.zip/

Table 1: Performance comparisons on Caltech-101

Num. of Samples	5	10	15	20	25	30
Wang [14]	51.1	59.7	65.4	67.7	70.1	73.4
ScSPM [15]	52.4	61.9	66.6	69.6	71.5	73.2
Ours	**53.1**	**62.2**	**67.0**	**69.9**	**72.2**	**73.7**



(a) Objective function values for training and testing (b) Performance with different dictionary size

Figure 1: Objective function values and classification performance

that a better dictionary can be learned by our algorithm, we analyze the objective function values, as shown in Eq.(1), for the training data and randomly sampled testing data, which is adopted as a criterion in [11, 12]. As can be seen from Fig.1(a), our algorithm returns a smaller object value than the ScSPM in both training and testing procedures, showing that a better dictionary and sparse representation of input data can be achieved by our method. In addition, we randomly selected 30 images per category as training data and performed our approach by using different dictionary size: 256, 512 and 1,024 respectively. The comparisons on the average classification accuracy with the ScSPM are shown in Fig.1(b), illustrating that our algorithm outperforms the ScSPM on both the small and the large codebook size.

We also compare our algorithm with the ScSPM on the 15 natural scenes dataset, which is widely used to verify the effectiveness for scene categorization task (Wang's algorithm has not been tested on this dataset). The average classification accuracy of our method is 81.6%, slightly higher than 80.7% achieved by the ScSPM, showing that our method also achieves good performance on scene categorization.

4. CONCLUSION AND FUTURE WORK

In this paper, we propose an efficient self-paced dictionary learning algorithm for image classification. Our approach introduces a better training order for dictionary learning. Since dictionary learning is a non-convex optimization problem, our algorithm can lead to a better locally optimal dictionary. Experimental results have shown that our algorithm consistently outperforms the baseline methods.

To further improve the classification performance, we are interested in developing a self-paced discriminative dictionary by combining the self-paced learning mechanism with supervised learning methods. Moreover, integrating the self-paced learning mechanism with online learning methods is also another meaningful issue.

5. ACKNOWLEDGMENTS

We would like to acknowledge the supports from the Program for New Century Excellent Talents of MOE China (Grant No. NCET-11-0213), the National 973 Program of China (Grant No. 2010CB327903), the National Science Foundation of China (Grant Nos. 61035003, 61021062, 60975043, 60875011), and the Key Program of National Science Foundation of Jiangsu, China (Grant Nos. BK2010054, BK2011005, BE2010638).

6. REFERENCES

[1] S. C. M. Bazara M S. *Nonlinear Programming Theory and Algorithms*. New York: John Wiley and Sons, 1979.

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. *In ICML*, pages 41–48, 2009.

[3] C.-F. Chen and Y.-C. F. Wang. Exploring self-similarities of bag-of-features for image classification. *In MM*, pages 1421–1424, 2011.

[4] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *In J. Construct. Approx*, pages 57–98, 1997.

[5] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *In Annals of Statistics*, pages 407–499, 2004.

[6] S. hua Zhong, Y. Liu, and Y. Liu. Bilinear deep learning for image classification. *In MM*, pages 343–352, 2011.

[7] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. *In CVPR*, pages 1697–1704, 2011.

[8] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. *In NIPS*, pages 1189–1197, 2010.

[9] M. P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. *In ICCV*, pages 1800–1807, 2011.

[10] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. *In NIPS*, pages 801–808, 2006.

[11] J. Mairal, F. Bach, and J. Ponce. Supervised translation-invariant sparse coding. *In TPMAI*, pages 791–804, 2012.

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. *In ICML*, pages 689–696, 2009.

[13] R. Tibshirani. Egression shrinkage and selection via the lasso. *In J. Royal Statistical*, pages 267–288, 1996.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *In CVPR*, pages 3360–3367, 2010.

[15] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *In CVPR*, pages 1794 – 1801, 2009.

[16] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. *In CVPR*, pages 3517–3524, 2010.

[17] H. Zhang, J. Yang, Y. Zhang, and T. Huang. Close the loop: Joint blind image restoration and recognition with sparse representation prior. *In ICCV*, pages 770–777, 2011.

[18] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *In J. Royal Statistical Soc*, pages 301–320, 2005.