# EFFICIENT SEMI-SUPERVISED ANNOTATION WITH PROXY-BASED LOCAL CONSISTENCY PROPAGATION

*Lei Huang*[1]*, Yang Wang*[2]*, Xianglong Liu*[1]*, Bo Lang*[1]

[1] State Key Laboratory of Software Development Environment, Beihang University, P.R.China
[2]National Computer Network Emergency Response Technical Team/Coordination Center of China
{huanglei, xlliu, langbo}@nlsde.buaa.edu.cn, aaron@ncic.ac.cn

## ABSTRACT

Semi-supervised learning methods can largely leverage the image annotation problem using both labeled and unlabeled data, especially when the labeled information is quite limited. However, most of them suffer the expensive computation stemming from the batch learning on large training dataset. In this paper we proposed a highly efficient semi-supervised annotation approach with the partial label propagation based on the graph representation. Specifically, the label information is first propagated from labeled samples to the unlabeled ones, and then spreads only among unlabeled ones like a spreading activation network. Our approach takes advantage of the decomposed formulation to achieve a fast incremental learning instead of the expensive batch one without accuracy loss. Extensive evaluations over two large datasets demonstrate the superior performance of the proposed method and its significant efficiency.

***Index Terms***— Image annotation, label propagation, semi-supervised learning, incremental learning

## 1. INTRODUCTION

Digital images have grown rapidly in recent years. The demand for effective solutions to manage images is increasing tremendously. Automatic image annotation is crucial to understanding image semantic concepts for storage, indexing and retrieval purposes. For most approaches of automatic image annotation, statistical models are usually built from manually labeled samples, and then the labels are assigned to unlabeled samples utilizing these models. However, this process faces a major problem that labeled data is often insufficient so that its distribution may not be able to well approximate that of the entire data set, which usually leads to inaccurate annotation results.

Semi-Supervised Learning (SSL) methods [1], by leveraging unlabeled data with certain assumptions, are promising to build more accurate models than those purely supervised methods. Typical SSL methods include self-training, co-training, transductive SVM, graph-based methods [2], etc. As an important family of SSL, graph-based methods have gained much attention in the past few years. Graph-based SSL define a graph which reflects the similarities among samples. It mainly involves two main components: graph construction and label propagation. Blum and Chawla [2] regard semi-supervised learning as a graph min-cut problem which is equivalent to the mode of a Markov random field with binary labels (Boltzmann machine). The Gaussian random fields and harmonic function method is a continuous relaxation to the discrete Markov random fields [3]. It can be viewed as a quadratic loss function with infinity weight and a regularizer based on the standard graph Laplacian [4]. The local and global consistency method [5] uses the normalized graph Laplacian and a classifying function sufficiently smooth with respect to the intrinsic structure revealed by the labeled and unlabeled points. In recent years, most of the literature focuses on studying the graph construction method [6, 7, 8]. Few works discuss the strategy of label propagation.

Our work mainly focuses on the strategy of label propagation and incremental learning. Different from the conventional methods that propagate the label information over the whole data set utilizing all effects among samples, we propose a novel label propagation algorithm named Proxy-based Local Consistency Propagation (PLCP). We assume the labeled samples should keep their labels unchanged [2, 3]. Thus the labeled samples shouldn't be affected mutually when label information propagates in the graph. Based on this intuition, we consider that the label information can be only propagated from labeled samples to unlabeled ones, namely the edges between them in the graph should be directed. Fig. 1 shows our propagation framework, where each labeled sample propagates the label information to its unlabeled neighbors (proxies), and then the proxies spread the information among the unlabeled samples mutually until a steady state is reached. Since each labeled data initially propagates its label to more than one unlabeled neighbors (proxies), our approach can be intuitively understood as increasing the labeled data

This work was supported by the National Major Research Plan of Infrastructure Software under Grant No.2010ZX01042-002-001-00 and the Foundation of the State Key Laboratory of Software Development Environment under Grant No.SKLSDE-2011ZX-01.

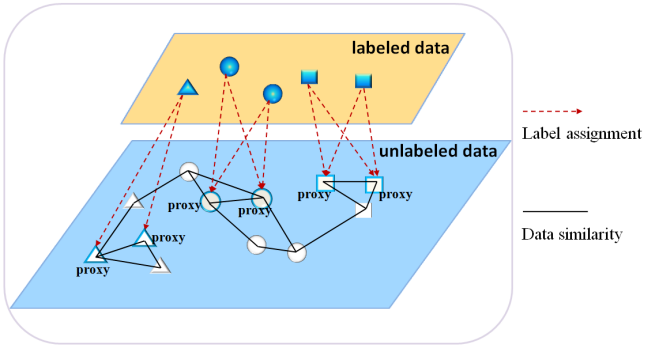


**Fig. 1**. The mixed graph constructed from data set with both labeled and unlabeled samples. The triangles, circles and squares represent three classes. The label information is first propagated from labeled samples to the proxies (regarded as a directed graph), and then spreads only among unlabeled ones (regarded as an undirected graph).

when performing propagation over the unlabeled data.

Many graph-based SSL methods have been applied to image or video annotation [9, 10, 11, 12]. However, most of them face the limitation that learning must be performed in a batch mode, which means that they require the training data set to be available all at once [13] and need a full retraining procedure when newly labeled samples emerge. Andrew et al. [14] propose an online learning algorithm for manifold regularization solved by a convex programming with stochastic gradient descent in kernel space. However it only achieves asymptotic zero-regret guarantee. Since our approach suppresses the mutual effect among labeled points, it can achieve fast incremental learning without accuracy loss through the decomposed formulation.

To summarize, we highlight here the main contributions of our work:

- We propose a novel label propagation algorithm named PLCP, in which the label information is first propagated from labeled samples to its unlabeled neighbors, and then spreads only among unlabeled ones like a spreading activation network [15].
- We propose an online semi-supervised framework and develop an incremental learning method for PLCP in which the newly added labeled samples can be efficiently used to update our annotation model.
- Our experiments show our algorithm has a promising performance and is quite efficient for online learning.

The remaining sections are organized as follows. Section 2 elaborates on the algorithm design and its interpretation. We propose a semi-supervised online learning framework and design the incremental learning algorithm for PLCP in Section 3. In Section 4, comprehensive experimental results are presented to demonstrate the effectiveness and efficiency of our approach. Finally Section 5 concludes this paper.

## 2. METHODOLOGY

First we describe the notation used in this paper. Given a point set $\chi = (\chi_L, \chi_U) = \{\mathbf{x}_1, ..., \mathbf{x}_l, \mathbf{x}_{l+1}, ..., \mathbf{x}_n\}$ where $\mathbf{x}_i \in R^d$. The first $l$ points $\chi_L = \{\mathbf{x}_1, ..., \mathbf{x}_l\}$ are labeled $y_i \in \mathcal{L} = \{1, ..., c\}$ and the remaining points $\chi_U = \{\mathbf{x}_{l+1}, ..., \mathbf{x}_n\}$ are unlabeled. The goal is to predict the label $y_j (l+1 \leq j \leq n)$ of the unlabeled points.

Let $F$ denote the set of matrices $n \times c$ with non-negative entries. A matrix $\mathbf{F} = [\mathbf{F}_1^T, ..., \mathbf{F}_n^T]^T \in F$ is a vectorial function $\mathbf{F} : \chi \rightarrow R^c$ which assigns a vector $\mathbf{F}_i$ to each point $\mathbf{x}_i$. The label matrix $\mathbf{Y} = [\mathbf{Y}_1^T, ..., \mathbf{Y}_n^T]^T$ is described as $\mathbf{Y} \in R^{n \times c}$ with $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ is with label $y_i = j$ and $\mathbf{Y}_{ij} = 0$ otherwise.

### 2.1. Algorithm

A typical assumption used in graph-based SSL is that nearby points are likely to have the same label. As the basis of label propagation, pairwise similarity measure is necessary for graph-based SSL methods. We use the pairwise similarity between samples as:

$$w_{ij} = exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}) \tag{1}$$

Our algorithm mainly focuses on the label propagation. We assume that the labeled data points shouldn't be affected mutually when label information propagates in the graph, so the label information can be only propagated directly from labeled points to unlabeled ones. Our algorithm can be described by two stages. At the first stage, the labeled samples view their neighbors as proxies which receive the label information. At the second stage, the proxies spread information among the unlabeled samples mutually until a steady state is reached.

**Stage** 1: The labels are propagated from the labeled points to the unlabeled ones. To reduce the propagation error, we propose to propagate the label information from each labeled point to its $K_{UL}$ nearest neighbors which are unlabeled. For each unlabeled point $\mathbf{x}_i \in \chi_U$, its initial label information $\mathbf{y}_i$ can be calculated by

$$\mathbf{y}_i = \sum_{j=1}^{l} \delta(\mathbf{x}_i \in N(\mathbf{x}_j)) w_{ij} \mathbf{y}_j \tag{2}$$

where $\delta(x)$ is the indicator function that returns 1 if x is true, otherwise 0. $\mathbf{x}_i \in N(\mathbf{x}_j)$ represents that $\mathbf{x}_i$ is one of the $K_{UL}$ nearest neighbors of $\mathbf{x}_j$. For convenience, we denote $\mathbf{Y}_L = [\mathbf{y}_1^T, ..., \mathbf{y}_l^T]^T$ , $\widehat{\mathbf{Y}}_U = [\mathbf{y}_{l+1}^T, ..., \mathbf{y}_n^T]^T$, so formula (2) can be expressed in a matrix form as:

$$\widehat{\mathbf{Y}}_U = \mathbf{W}_{UL} \mathbf{Y}_L \tag{3}$$

where $\mathbf{W}_{UL} \in R^{u \times l}$, whose entries $w_{ij}$ can be calculated by formula (1) if $\mathbf{x}_i \in N(\mathbf{x}_j)$ otherwise 0. $u$ is the size of unlabeled data set and $u + l = n$.

**Stage 2**: The label information spreads only among unlabeled points. At first, we form the affinity matrix $\mathbf{W}_U$ whose entries $w_{ij}$ can be calculated according to formula (1) if $(i \neq j)$ otherwise 0. Then we construct the matrix $\mathbf{S}_U = \mathbf{D}_U^{-\frac{1}{2}}\mathbf{W}_U\mathbf{D}_U^{-\frac{1}{2}}$, in which $\mathbf{D}_U$ is a diagonal matrix with its (*i*, *i*)-element equal to the sum of the *i*-th row of $\mathbf{W}_U$. Similar to $\mathbf{F}$, we denote $\mathbf{F}_U = [(\mathbf{F}_U^1)^T, ..., (\mathbf{F}_U^u)^T]^T \in F_U$, which is a vectorial function $\mathbf{F}_U : \chi_U \to R^c$, and then we can calculate it iteratively as follows:

$$\mathbf{F}_U(t+1) = \alpha\mathbf{S}_{UU}\mathbf{F}_U(t) + (1-\alpha)\widehat{\mathbf{Y}}_U \tag{4}$$

where $\alpha$ is a parameter in (0, 1). Formula (4) can be understood intuitively in terms of spreading activation networks where each unlabeled point receives the label information from its unlabeled neighbors (first term), and also receives the label information from labeled ones (second term) which can be regarded as external inputs.

Let $\mathbf{F}^*$ be the limit of the sequence $\{\mathbf{F}_U(t)\}$. We can assign each point $\mathbf{x}_i \in \chi_U$ with label $y_i = \arg\max_{j \leq c} \mathbf{F}^*_{ij}$ .

Following previous work [5], it is easy to show that sequence $\{\mathbf{F}_U(t)\}$ converges with $\mathbf{F}_U(0) = \widehat{\mathbf{Y}}_U$. Hence

$$\begin{aligned}\mathbf{F}^* &= \lim_{t\to\infty} \mathbf{F}_U(t) = (1-\alpha)(\mathbf{I} - \alpha\mathbf{S}_U)^{-1}\widehat{\mathbf{Y}}_U \\ &= (1-\alpha)(\mathbf{I} - \alpha\mathbf{S}_U)^{-1}\mathbf{W}_{UL}\mathbf{Y}_L \end{aligned} \tag{5}$$

Our approach can be represented as a mixed graph $G = \langle \chi, E \rangle$, which consists of a set of vertices, denoted by $\chi = (\chi_L, \chi_U)$ and a set of edges $E = E_{LU} \bigcup E_{UU}$, where $E_{LU} \subseteq \chi_L \times \chi_U$ is a set of directed edges, $E_{UU} \subseteq \chi_U \times \chi_U$ is a set of undirected edges. The label information is first propagated from labeled samples to the unlabeled ones, and then spreads only among unlabeled ones like a spreading activation network, as shown in Fig. 1.

### 2.2. Formulation and interpretation

Here we develop a regularization framework for the above iteration algorithm. The cost function associated with $\mathbf{F}_U$ is defined to be

$$\begin{aligned}Q(\mathbf{F}_U) = &\frac{1}{2}\sum_{i,j=1}^{u} w_{ij}\|\frac{\mathbf{F}_U^i}{\sqrt{\mathbf{D}_U^i}} - \frac{\mathbf{F}_U^j}{\sqrt{\mathbf{D}_U^j}}\|^2 \\ &+\frac{\mu}{2}\sum_{i=1}^{u}\|\mathbf{F}_U^i - [\mathbf{W}_{UL}\mathbf{Y}_L]_i\| \end{aligned} \tag{6}$$

where $[\mathbf{M}]_i$ represents the *i*-th row vector of matrix $\mathbf{M}$ and $\mu > 0$ is the regularization parameter. $\mathbf{D}_U^i$ represents the (*i*, *i*)-element of $\mathbf{D}_U$. Then the classifying function is

$$\mathbf{F}^* = \arg\min_{\mathbf{F}_U \in F_U} Q(\mathbf{F}_U) \tag{7}$$

intuitively, the first term of $Q(\mathbf{F}_U)$ is the smoothness constraint, which means that a good classifying function for unlabeled points should not change too much between nearby points. The second term is the fitting constraint, which means a good classifying function for unlabeled points should not change too much from the assignment of the labeled points. The trade-off between these two competing constraints is captured by a positive parameter $\mu$.

Differentiating $Q(\mathbf{F}_U)$ with respect to $\mathbf{F}_U$, we have

$$\frac{\partial Q}{\partial \mathbf{F}_U} = \mathbf{F}_U - \mathbf{S}_U\mathbf{F}_U + \mu(\mathbf{F}_U - \mathbf{W}_{UL}\mathbf{Y}_L)$$

Because $Q(\mathbf{F}_U)$ is convex, we can get $\mathbf{F}^*$ by solving $\frac{\partial Q}{\partial \mathbf{F}_U} = 0$. So we have

$$\mathbf{F}^* - \mathbf{S}_U\mathbf{F}^* + \mu(\mathbf{F}^* - \mathbf{W}_{UL}\mathbf{Y}_L) = 0$$

which can be transformed into

$$(\mathbf{I} - \alpha\mathbf{S}_U)\mathbf{F}^* = (1-\alpha)\mathbf{W}_{UL}\mathbf{Y}_L$$

where $\alpha = \frac{1}{1+\mu}$. Since $(\mathbf{I} - \alpha\mathbf{S}_U)$ is invertible, we have

$$\mathbf{F}^* = (1-\alpha)(\mathbf{I} - \alpha\mathbf{S}_U)^{-1}\mathbf{W}_{UL}\mathbf{Y}_L \tag{8}$$

which recovers the same expression of formula (5).

### 2.3. Related works and discussions

It is worth noting that our approach is similar to Gaussian Fields and Harmonic Functions (GFHF) [3], in which the graph regularizer based loss function is defined as:

$$Q(\mathbf{F}) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\|\mathbf{F}_i - \mathbf{F}_j\|^2 + \infty\sum_{i=1}^{l}\|\mathbf{F}_i - \mathbf{Y}_i\|^2$$

where $\mathbf{Y}_i$ is the *i*-th row vector of $\mathbf{Y}$. However there are some differences between GFHF and our approach. GFHF uses standard graph Laplacian as the smoothness regularizer while our approach uses the normalized one. GFHF constructs the undirected graph to propagate mutually over the whole data set. However when computing the harmonic solution, GFHF fixes the labeled points with the given label values. Our approach uses a directed propagation from labeled data to unlabeled ones when fixing the labeled points.

Our approach is also similar but different to Local and Global Consistency (LGC) [5], whose regularizer is proposed as:

$$Q(\mathbf{F}) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\|\frac{\mathbf{F}_i}{\sqrt{\mathbf{D}_i}} - \frac{\mathbf{F}_j}{\sqrt{\mathbf{D}_j}}\|^2 + \frac{\mu}{2}\sum_{i=1}^{n}\|\mathbf{F}_i - \mathbf{Y}_i\|$$

where $\mathbf{D} = diag(d_i)$ defined as $d_i = \sum_{j=1}^{n} w_{ij}$. LGC doesn't fix the labeled points with the given label values and allow them be changed by using the normalized Laplacian among all data. It considers the mutual effect among labeled points while our approach suppresses the negative effect. Our approach fixes the labeled points and propagates the label information from labeled points to unlabeled ones.

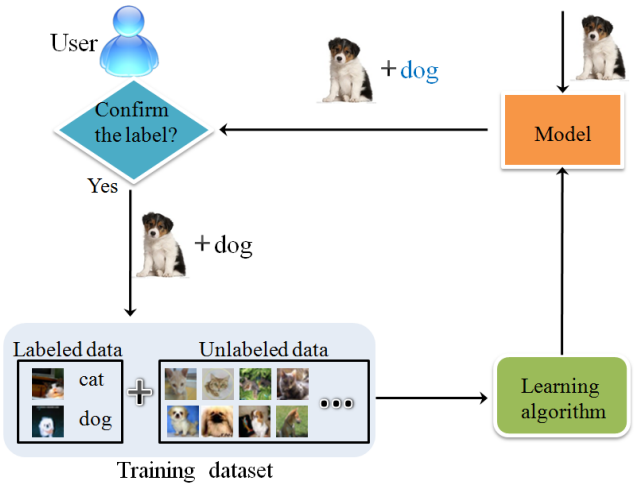


**Fig. 2**. Illustration of our online semi-supervised annotation framework. Initially the model is learned using labeled and unlabeled data and we can generate the label of the unlabeled data and the predictor. For a new image, we can predict its label through the predictor. If the user confirms the label of the image, we can add the new data to the training data set and retrain the model, which updates the label of the unlabeled data and the predictor.

## 3. INCREMENTAL LEARNING

In this section, we propose an online learning framework and develop an incremental learning algorithm for PLCP.

### 3.1. Online semi-supervised annotation framework

We propose an online semi-supervised annotation framework for image annotation (Fig. 2):

1. Given some labeled data points $\{\chi_L, \mathbf{Y}_L\}$ and large scale unlabeled points $\chi_U$, the model is learned utilizing both labeled and unlabeled data in a semi-supervised manner. So we can get the predicted labels $\mathbf{F}_U$ of the unlabeled data $\chi_U$ initially.
2. For a new image $\mathbf{x}_{new}$ the system makes prediction $\mathbf{f}_{new}$ using its current predictor and shows the prediction to the user.
3. If the user confirms the label $\mathbf{y}_{new}$, $(\mathbf{x}_{new}, \mathbf{y}_{new})$ should be treated as training data to retrain the model, which can update the label information of the unlabeled data and the predictor.

Different from Andrew's work [14], which uses both the newly labeled and unlabeled data, our proposed online semi-supervised annotation framework only relies on labeled data in online updating process. Most SSL methods learn in a batch mode, which can not satisfy the requirement of online real-time annotation [13]. Some may support learning in an incremental mode, but suffer from accuracy loss [14]. However, our approach PLCP doesn't lose accuracy in online updating process by taking advantage of the decomposed formulation. Next we will elaborate on it.

### 3.2. Incremental learning

For convenience, we denote $\mathbf{T}_U = (1-\alpha)(\mathbf{I}-\alpha\mathbf{S}_U)^{-1}$. Given a new labeled data point $(\mathbf{x}_{new}, \mathbf{y}_{new})$, we add it to the training data set and retrain the model. For $\mathbf{x}_i \in \chi_U$, its new predicted label information $\mathbf{F}'_U$ can be calculated by :

$$\begin{aligned}\mathbf{F}'_U &= \mathbf{T}_U[\mathbf{W}_{UL}, W_{U,new}] \cdot [\mathbf{Y}_L^T, \mathbf{y}_{new}^T]^T \\ &= \mathbf{F}_U + \mathbf{T}_U W_{U,new}\mathbf{y}_{new}\end{aligned} \tag{9}$$

where $W_{U,new} = \{w_{l+1,new}, ...w_{n,new}\}$, and $w_{i,new}(l+1 \leq i \leq n)$can be calculated by formula (1) if $\mathbf{x}_i \in N(\mathbf{x}_{new})$ otherwise 0. It is worth noting that $\mathbf{T}_U$ is a constant and we can calculate and store it in the initial step. We can decompose formula (9) as a more concise form. We denote $\Delta = W_{U,new}\mathbf{y}_{new}$ and $\Delta \in R^{n\times c}$. For multi-class annotation, we know $\mathbf{y}_{new} \in (0,1)^c$ and only one nonzero element $\mathbf{y}_{new}^{(j)} = 1(1 \leq j \leq c)$ exists which means $\mathbf{x}_{new}$ only has label *j*. So for $\Delta$, only the *j*-th column is nonzero. We denote it as $\Delta_{\cdot j}$. Then we only need to update the *j*-th column of $\mathbf{F}_U$ with an increment $\Delta\mathbf{F}_U^j = \mathbf{T}_U\Delta_{\cdot j}$.

For incrementally adding *m* new labeled data $(\chi_M, \mathbf{Y}_M)$, we can get the similar result:

$$\mathbf{F}'_U = \mathbf{F}_U + \mathbf{T}_U\mathbf{W}_{UM}\mathbf{Y}_M \tag{10}$$

where $\mathbf{W}_{UM} \in R^{u\times m}$, whose entries $w_{ij}$ can be calculated by formula (1) if $\mathbf{x}_i \in N(\mathbf{x}_j)$ otherwise 0.

Based on $\Delta\mathbf{F}_U^j = \mathbf{T}_U\Delta_{\cdot j}$, we can know that the time complexity is $O(m \times u^2)$, which is lineal to *m* and greatly enhances the efficiency.

## 4. EXPERIMENT

In this section, we give a set of experiments to evaluate our approach. We validate the effectiveness of our PLCP for semi-supervised classification, including digits recognition and image annotation, compare to *k* nearest neighbor (*k*NN) algorithm, Gaussian Fields and Harmonic Functions (GFHF) [3] and Local and Global Consistency (LGC) [5]. We also evaluate the efficiency of PLCP in an online mode. In all experiments, we evaluate the transductive accuracy on unlabeled data and we report the averaged performance of 10 runs to suppress the randomness.

### 4.1. Digits recognition

We adopt MNIST [1] consisting of 70k $28 \times 28$ handwritten digits images for digits recognition. Considering the limitation of both computing ability and memory quantity, we randomly sampled 10k samples as our data set. Each image is

[1] http://yann.lecun.com/exdb/mnist/

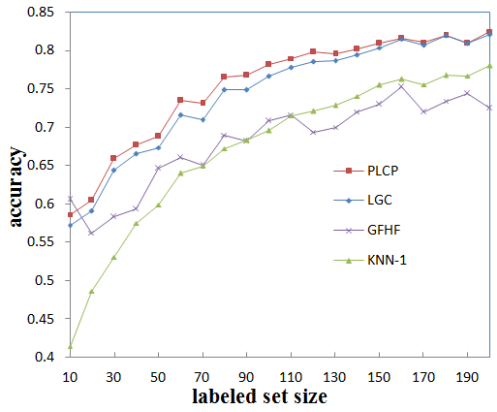


**Fig. 3**. The accuracy of digit recognition with MNIST handwritten digits dataset. The size of unlabeled data set is 10k.

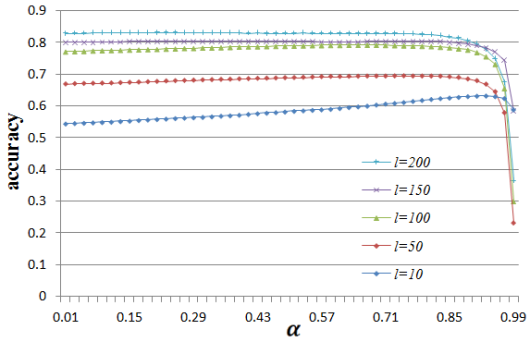


**Fig. 4**. Robustness to different values of  in PLCP. The horizontal axis represents the different value of , and the vertical axis is the total recognition accuracy value. The size of randomly labeled data set is set to 10, 50, 100, 150, 200 respectively.

represented by a 784 dimensional vector with the raw values ranging from 0 to 255.

We compared our method to *k*NN, GFHF coupled with the Class Mass Normalization (CMN) and LGC. The $k$ in *k*NN was set to 1, which is the optimal value according to our observation. In our method and LGC, the affinity matrix was constructed by using *k*NN with $k$ set to 100 empirically. The width of RBF kernel was set to the max value of the pairwise distances and the diagonal elements of affinity matrix were set to 0. The value of parameter $\alpha$ was simply selected from the optimal value of range (0.09, 0.99) with interval 0.1. $K_{UL}$ was empirically set to 40. It should be noticed that for GFHF, since the classification result is very sensitive to the width of RBF kernel, we selected the optimal value after fine tuning.

Fig. 3 shows the results comparing our method with three baselines. It can be observed that (1) all methods increase their performances with more labeled samples; (2) our PLCP achieved better accuracy than all the other methods, especially when 20-150 samples were available.

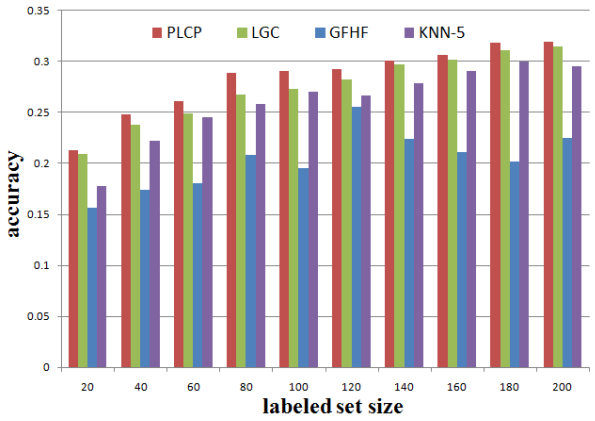


**Fig. 5**. The accuracy of image annotation with CIFAR-10 dataset for 5,000 test data sampled.

We also studied the sensitivity of parameter $\alpha$ in PLCP. The results are shown in Fig. 4. We can find that the accuracy maintains a stable value with a little disturbance when $\alpha \in (0.01, 0.91)$. For $\alpha \in (0.91, 0.99)$), the accuracy reduces dramatically. This is because that the unlabeled sample receives more label information from the unlabeled ones in PLCP. Thus we can find that as long as the ratio of the label information from the labeled samples is not too small (for example, $1-\alpha \geq 0.1$)), the PLCP has good and stable results.

### 4.2. Image annotation

In this experiment, we investigated the task of image annotation using the CIFAR-10 data set [2] , which consists of 60k $32 \times 32$ color images in 10 classes, such as airplane, automobile, bird, etc. We extracted 384 dimensional GIST feature for each image [16]. To build the test data set, we randomly sampled 5000 data points from the whole data set.

Fig. 5 shows the results comparing our method to *k*NN, GFHF and LGC. In *k*NN, we set the optimal value *k*=5. The affinity matrix was constructed in a way similar to that of Section 4.1. The value of parameter $\alpha$ was set to 0.19. From Fig. 5, we can find that our PLCP outperforms all the other methods significantly. For all labeled set size, the mean accuracy of PLCP has an improvement of $9.57\%$, $40.59\%$ and $3.62\%$ over *k*NN, GFHF and LGC respectively. We can find that GFHF has a poor performance in this dataset. The likely reason is that the data points don't live on a single manifold and GFHF is very sensitive to it.

### 4.3. Incremental learning

In this experiment, we compare the computation time of Incremental PLCP (PLCP-INC) with PLCP, GFHF and LGC.

[2] http://www.cs.toronto.edu/ kriz/cifar.html

**Table 1**. Computational cost of GFHF, LGC, PLCP and PLCP-INC in the online annotation mode.

| k-th round | PLCP time(s) | GFHF time(s) | LGC time(s) | PLCP-INC time(s) |
|---|---|---|---|---|
| k=1 | 17.30 | 8.48 | 21.83 | 0.0605 |
| k=2 | 17.61 | 10.10 | 27.20 | 0.0640 |
| k=3 | 17.79 | 11.50 | 34.14 | 0.0572 |
| k=4 | 18.16 | 13.18 | 42.08 | 0.0595 |
| k=5 | 18.28 | 15.45 | 51.02 | 0.0588 |
| k=6 | 18.36 | 19.22 | 60.92 | 0.0651 |
| k=7 | 18.65 | 21.40 | 72.49 | 0.0588 |
| k=8 | 19.11 | 24.12 | 85.22 | 0.0609 |
| k=9 | 19.78 | 28.11 | 99.53 | 0.0645 |
| k=10 | 20.07 | 31.57 | 115.17 | 0.0589 |

The difference between PLCP-INC and PLCP should be noticed, namely PLCP-INC utilizes the formulation decomposition as described in Section 3.2 when the model is retrained and PLCP doesn't. We implemented all methods using MATLAB in a 2.13GHz server. All data were sampled from MNIST dataset. It is worth noting that our PLCP-INC doesn't lose accuracy compared with PLCP, so here we omit the accuracy comparison.

The size of test set (unlabeled data) is 5000. All parameter were set to the same value as Section 4.1. We simulated the online annotation procedure by incrementally providing the labeled training data as a sequence $\{\Delta l_i, i = 1, 2, ..., T\}$. That is to say, at the *k*-th round, the newly labeled data $\Delta l_k$ is provided, $\Delta l_k$ and all the previous labeled data $\{\Delta l_i, i = 1, 2, ..., k-1\}$ will be treated as the new training set. We set the size of incremental labeled data $|\Delta l_i = 500|$ and the total number of rounds T=10.

The computational cost of all compared methods is listed in Table 1. (PLCP-INC initially needs to calculate $\mathbf{T}_U$ on the test set, which cost $t_0 = 8s$ in this experiment). It should be noticed that the computational time includes the time of graph construction and label propagation. From Table 1, we can see that PLCP-INC significantly reduced the computation burden compared to PLCP, GFHF and LGC. With the round *k* increases, the computational cost of all baselines increases sharply. In contrast, PLCP-INC remains very stable around 0.06s, which satisfies the requirement of online real-time annotation.

## 5. CONCLUSION

In this paper, we proposed a novel graph based SSL approach named Proxy-based Local Consistency Propagation (PLCP), which divides the label propagation into two stages. Based on PLCP, we have developed an incremental learning algorithm that uses the newly added labeled samples to efficiently update the learned model. Our experiments show that our algorithm has a promising performance and can satisfy the requirement of online real-time annotation.

## 6. REFERENCES


[1] X. Zhu, "Semi-Supervised Learning Literature Survey," 2005.

[2] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *ICML*, 2001.

[3] X. Zhu, Z. Ghahramani, and J. D. Laffertyz, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NIPS*, 2002.

[5] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *NIPS*, 2004.

[6] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *ICML*, 2006.

[7] M. Ghazvininejad, M. Mahdieh, H. R. Rabiee, P. K. Roshan, and M. H. Rohban, "Isograph: neighbourhood graph construction based on geodesic distance for semi-supervised learning," in *ICDM*, 2011.

[8] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *ICML*, 2009.

[9] J. Tang, X.-S. Hua, G.-J. Qi, M. Wang, T. Mei, and X. Wu, "Structure-sensitive manifold ranking for video concept detection," in *ACM Multimedia*, 2007.

[10] J. Wang, S.-F. Chang, X. Zhou, and S.T. Wong, "Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels," in *CVPR*, 2008.

[11] J. Tang, R. Hong, S. Yan, T.-S Chua, G.-J Qi, and R. Jain, "Image annotation by knn-sparse graph-based label propagation over noisily-tagged web images," *ACM TIST*, vol. 2, no. 2, pp. 1–15, 2011.

[12] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu, "A generic framework for video annotation via semi-supervised learning," *IEEE Transaction. Multimedia*, vol. 14, no. 4, pp. 1206–1219, Aug. 2012.

[13] X. Zhu, A. B. Goldberg, and T. Khot, "Some new directions in graph-based semi-supervised learning," in *ICME*, 2009.

[14] A. B. Goldberg, M. Li, and X. Zhu, "Online manifold regularization: a new learning setting and empirical study," in *ECML*, 2008.

[15] J. Shrager, T. Hogg, and B. A. Huberman, "Observation of phase transitions in spreading activation networks," *Science*, vol. 236, no. 4805, May. 1987.

[16] X. Liu, Y. Mu, B. Lang, and S.-F. Chang, "Compact hashing for mixed image-keyword query over multi-label images," in *ICMR*, 2012.