IMAGE CLASSIFIER LEARNING FROM NOISY LABELS VIA GENERALIZED GRAPH SMOOTHNESS PRIORS

Yu Mao [*], *Gene Cheung* [#], *Chia-Wen Lin* [$], *Yusheng Ji* [#]

[*] The Graduate University for Advanced Studies, [#] National Institute of Informatics,
[$] National Tsing Hua University

ABSTRACT

When collecting samples via crowd-sourcing for semi-supervised learning, often labels that designate events of interest are assigned unreliably, resulting in label noise. In this paper, we propose a robust method for graph-based image classifier learning given noisy labels, leveraging on recent advances in graph signal processing. In particular, we formulate a graph-signal restoration problem, where the objective includes a fidelity term to minimize the l_0-norm between the observed labels and a reconstructed graph-signal, and generalized graph smoothness priors, where we assume that the reconstructed signal and its gradient are both smooth with respect to a graph. The optimization problem can be efficiently solved via an iterative reweighted least square (IRLS) algorithm. Simulation results show that for two image datasets with varying amounts of label noise, our proposed algorithm outperforms both regular SVM and a noisy-label learning approach in the literature noticeably.

Index Terms— Graph-based classifiers, label denoising, generalized smoothness priors

1. INTRODUCTION

The prevalence of social media sites like Facebook and Instagram means that *user-generated content* (UGC) like selfies is growing rapidly. Classification of this vast content into meaningful categories can greatly improve understanding and detect prevailing trends. However, the sheer size of UGC means that it is too costly to hire experts to assign labels (classification into different events of interest) to partial data for semi-supervised classifier learning.

One approach to this big data problem is *crowd-sourcing* [1]: employ many non-experts online to assign labels to a subset of data at a very low cost. However, non-experts can often be unreliable (*e.g.*, a non-expert is not competent in a label assignment task but pretends to be, or he simply assigns label randomly to minimize mental effort), leading to label errors or noise.

In this paper, we propose a new method to robustly learn a graph-based image classifier given (partial) noisy labels, leveraging on recent advances in graph signal processing (GSP) [2]. In particular, we formulate a graph-signal restoration problem, where the graph-signal is the desired labeling of all samples into two events. The optimization objective includes a fidelity term that measures the l_0-norm between the observed labels in the training samples and the reconstructed signal, and generalized graph smoothness priors that assume the desired signal *and* its gradient are smooth with respect to a graph—an extension of *total generalized variation* (TGV) [3] to the graph-signal domain. Because the notion of smoothness applies to the entire graph-signal, unlike SVM that considers only samples along boundaries that divide the feature space into clusters of different events, all available samples are considered during graph-signal reconstruction, leading to a more robust classifier. The optimization is solved efficiently via an *iterative reweighted least square* (IRLS) algorithm [4]. Simulation results for two image datasets with varying amount of label noise show that our proposed algorithm outperforms both regular SVM and a noisy-label learning approach in the literature noticeably.

The outline of the paper is as follows. We first overview related works in Section 2. We then review basic GSP concepts and define graph smoothness notions in Section 3. In Section 4, we describe our graph construction using available samples and formulate our noisy label classifier learning problem. We present our proposed IRLS algorithm in Section 5. Finally, we present experimental results and conclusions in Section 6 and 7, respectively.

2. RELATED WORK

Learning with label noise has garnered much interest, including a workshop[1] in NIPS'10 [1] and a journal special issue in Neurocomputing [5]. There exists a wide range of approaches, including theoretical (*e.g.*, label propagation in [6]) and application-specific (*e.g.* emotion detection using inference algorithm based on multiplicative update rule [7]). In this paper, similar to previous works [8, 9, 10, 11] we choose to build a *graph-based classifier*, where each acquired sample is represented as a node in a high-dimensional feature space and connects to other sample nodes in its neighborhood. Our approach is novel in that we show how generalized graph smoothness notion—extending TGV [3] to the graph-signal domain—can be used for robust graph-based classifier learning with label noise.

Graph-signal priors have been used for image restoration problems such as denoising [12, 13, 14], interpolation [15, 16], bit-depth enhancement [17] and JPEG de-quantization [18]. The common assumption is that the desired graph-signal is smooth or band-limited with respect to a properly chosen graph that reflects the structure of the signal. In contrast, we define a generalized notion of graph smoothness for signal restoration specifically for classifier learning.

3. SMOOTHNESS OF GRAPH-SIGNALS

3.1. Preliminaries

GSP is the study of signals on structured data kernels described by graphs [2]. We focus on undirected graphs with non-negative edge weights. A weighted undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ consists of a finite set of vertices $\mathcal{V}$ with cardinality $|\mathcal{V}| = N$, a set of edges $\mathcal{E}$ connecting vertices, and a weighted adjacency matrix $\mathbf{W}$. $\mathbf{W}$ is a real $N \times N$ symmetric matrix, where $w_{i,j} \geq 0$ is the weight assigned to the edge (i, j) connecting vertices i and j, $i \neq j$.

Given $\mathcal{G}$, the *degree matrix* $\mathbf{D}$ is a diagonal matrix whose i-th diagonal element $D_{i,i} = \Sigma_{j=1}^{N} w_{i,j}$. The *combinatorial graph Laplacian* $\mathbf{L}$ (graph Laplacian for short) is then:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{1}$$

[1] https://people.cs.umass.edu/ wallach/workshops/nips2010css/

978-1-5090-0746-2/16/$31.00

Because $\mathbf{L}$ is a real symmetric matrix, there exists a set of eigenvectors ϕ_i with corresponding real eigenvalues λ_i that decompose $\mathbf{L}$, *i.e.*,

$$\mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T = \sum_i \lambda_i \phi_i \phi_i^T = \mathbf{L} \tag{2}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues λ_i on its diagonal, and $\mathbf{\Phi}$ is an eigenvector matrix with corresponding eigenvectors ϕ_i as its columns. $\mathbf{L}$ is positive semi-definite [2], *i.e.* $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$, $\forall \mathbf{x} \in \mathbb{R}^N$, which implies that the eigenvalues are non-negative, *i.e.* $\lambda_i \geq 0$. The eigenvalues can be interpreted as frequencies of the graph. Hence any signal $\mathbf{x}$ can be decomposed into its graph frequency components via $\mathbf{\Phi}^T\mathbf{x}$, where $\alpha_i = \phi_i^T\mathbf{x}$ is the i-th frequency coefficient. $\mathbf{\Phi}^T$ is called the *graph Fourier transform* (GFT).

3.2. Generalized Graph Smoothness

We next define the notion of "smoothness" for graph-signals. $\mathbf{x}^T\mathbf{L}\mathbf{x}$ captures the total variation of signal $\mathbf{x}$ with respect to graph $\mathcal{G}$ in l_2-norm:

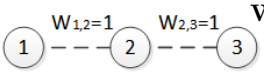
$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{i,j}\,(x_i - x_j)^2 \tag{3}$$

In words, $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is small if connected vertices x_i and x_j have similar signal values for edge $(i,j) \in \mathcal{E}$, *or* if the edge weight $w_{i,j}$ is small.

$\mathbf{x}^T\mathbf{L}\mathbf{x}$ can also be expressed in terms of graph frequencies λ_i:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \left(\mathbf{x}^T\mathbf{\Phi}\right)\mathbf{\Lambda}\left(\mathbf{\Phi}^T\mathbf{x}\right) = \sum_i \lambda_i \alpha_i^2 \tag{4}$$

Thus a small $\mathbf{x}^T\mathbf{L}\mathbf{x}$ also means that the energy of signal $\mathbf{x}$ is mostly concentrated in the low graph frequencies.



$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) line graph (b) adjacency and degree matrices

Fig. 1. Example of a line graph with three nodes and edge weights 1, and the corresponding adjacency and degree matrices $\mathbf{W}$ and $\mathbf{D}$.

Like TGV, we can also define a higher-order notion of smoothness. Specifically, $\mathbf{L}$ is related to the second derivative of continuous functions [2], and so $\mathbf{L}\mathbf{x}$ computes the second-order difference on graph-signal $\mathbf{x}$. As an illustrative example, a 3-node line graph with edge weight $w_{i,j} = 1$, shown in Fig. 1, has the following graph Laplacian:

$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \tag{5}$$

Using the second row $\mathbf{L}_{2,:}$ of $\mathbf{L}$, we can compute the second-order difference at node x_2:

$$\mathbf{L}_{2,:}\mathbf{x} = -x_1 + 2x_2 - x_3 \tag{6}$$

On the other hand, the definition of second derivative[2] of a function $f(x)$ is:

$$f''(x) = \lim_{h\to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{7}$$

We see that (6) and (7) are computing the same quantity in the limit.

[2]https://en.wikipedia.org/wiki/Second_derivative

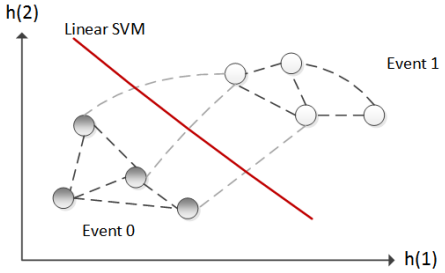


Fig. 2. Example of a constructed graph $\mathcal{G}$ for binary-event classification with two features $h(1)$ and $h(2)$. A linear SVM would dissect the space into two for classification.

Hence if $\mathbf{L}\mathbf{x}$ is small, then the second-order difference of $\mathbf{x}$ is small, or the first-order difference of $\mathbf{x}$ is smooth or changing slowly. In other words, the *gradient* of the signal is smooth with respect to the graph. We express this notion by stating that the square of the l_2-norm of $\mathbf{L}\mathbf{x}$ is small:

$$\|\mathbf{L}\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{L}^T\mathbf{L}\mathbf{x} = \mathbf{x}^T\mathbf{L}^2\mathbf{x} = \sum_i \lambda_i^2\alpha_i^2 \tag{8}$$

where (8) is true since $\mathbf{L}$ is symmetric by definition.

4. PROBLEM FORMULATION

4.1. Graph Construction

In a semi-supervised learning scenario, we assume that a set of training samples of size N_1 with possibly noisy *binary* labels (*i.e.*, two-event classification) are available, and we are tasked to classify N_2 additional test samples. Denote by $\mathbf{x}$ the length-N vector of ground truth binary labels, where $x_i \in \{-1, 1\}$ and $N = N_1 + N_2$. Similarly, denote by $\mathbf{y}$ the length-N_1 vector of observed training labels, where $y_i \in \{-1, 1\}$.

We first construct a graph to represent all N samples. Each sample is represented as a vertex on the graph $\mathcal{G}$ and has an associated set of M features $h_i(m)$, $1 \leq m \leq M$. Given available features, we can measure the *similarity* between two samples i and j and compute the edge weight $w_{i,j}$ between vertices i and j in the graph $\mathcal{G}$ as follows:

$$w_{i,j} = \exp\left(\frac{-\sum_{m=1}^{M} c_m\,(h_i(m) - h_j(m))^2}{\sigma_h^2}\right) \tag{9}$$

where σ_h is a parameter and c_m is a *correlation factor* that evaluates the correlation between feature $h_i(m)$ of N_1 samples and the samples' labels y_i. In words, (9) states that two sample vertices i and j has edge weight $w_{i,j}$ close to 1 if their associated *relevant* features are similar, and close to 0 if their relevant features are different. This method of graph construction is very similar to previous works on graph-based classifiers [8, 9, 10, 11].

For robustness, we connect all vertex pairs i and j in the vertex set $\mathcal{V}$, resulting in a complete graph. Empirical results show that a more connected graph is more robust to noise than a sparse graph. An example constructed graph is shown in Fig. 2, where each sample i has two features $h_i(1)$ and $h_i(2)$. Two samples i and j with similar relevant features will have a small distance in the feature space and an edge weight $w_{i,j}$ close to 1. A linear SVM would divide the feature space into two halves for a two-event classification. Having defined edge weights $w_{i,j}$, the graph Laplacian $\mathbf{L}$ can be computed as described in Section 3.1.

4.2. Label Noise Model

To model label noise, we adopt a uniform noise model [1], where the probability of observing $y_i = x_i$, $1 \leq i \leq N_1$, is $1-p$, and p otherwise; *i.e.*,

$$Pr(y_i|x_i) = \begin{cases} 1-p & \text{if } y_i = x_i \\ p & \text{o.w.} \end{cases} \tag{10}$$

Hence the probability of observing a noise-corrupted $\mathbf{y}$ given ground truth $\mathbf{x}$ is:

$$\begin{aligned} Pr(\mathbf{y}|\mathbf{x}) &= p^k (1-p)^{N_1 - k} \\ k &= \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_0 \end{aligned} \tag{11}$$

where $\mathbf{D}$ is a $N_1 \times N$ binary matrix that selects the first N_1 entries from length-N vector $\mathbf{x}$. (11) serves as the *likelihood* or *fidelity* term in our MAP formulation.

4.3. Graph-Signal Prior

For signal prior $Pr(\mathbf{x})$, following the discussion in Section 3.2 we assume that the desired signal $\mathbf{x}$ *and* its gradient are smooth with respect to a graph $\mathcal{G}$ with graph Laplacian $\mathbf{L}$. Mathematically, we use the Gaussian kernel to define $Pr(\mathbf{x})$:

$$Pr(\mathbf{x}) = \exp\left(-\frac{\mathbf{x}^T \mathbf{L}\, \mathbf{x}}{\sigma_0^2}\right) \exp\left(-\frac{\mathbf{x}^T \mathbf{L}^2\, \mathbf{x}}{\sigma_1^2}\right) \tag{12}$$

where σ_0 and σ_1 are parameters. One can interpret (12) as an extension of TGV [3] to the graph-signal domain.

We interpret the two smoothness terms in the context of binary-event classification. We know that the ground truth signal $\mathbf{x}$ is indeed *piecewise smooth*; each true label x_i is binary, and labels of the same event cluster together in the same feature space area. The signal smoothness term in (12) promotes piecewise smoothness in the reconstructed graph-signal $\hat{\mathbf{x}}$, as shown in previous graph-signal restoration works [13, 14, 18], and hence is an appropriate prior here.

Recall that the purpose of TGV [3] is to avoid over-smoothing a *ramp* (linear increase / decrease in pixel intensity) in an image, which would happen if only a total variation (TV) prior is used. A ramp in the reconstructed signal $\hat{\mathbf{x}}$ in our classification context would mean an assignment of label other than -1 and 1, which can reflect the *confidence level* in the estimated label; *e.g.*, a computed label $\hat{x}_i = 0.3$ would mean the classifier has determined that event i is more likely to be 1 than -1, but the confidence level is not high. *By using the gradient smoothness prior, one can promote the appropriate amount of ambiguity in the classification solution instead of forcing the classifier to make hard binary decisions.* As a result, the mean square error (MSE) of our solution with respect to the ground truth labels is low.

4.4. Objective Function

We can now combine the likelihood and signal prior together to define an optimization objective. Instead of maximizing the posterior probability $Pr(\mathbf{x}|\mathbf{y}) \propto Pr(\mathbf{y}|\mathbf{x})Pr(\mathbf{x})$, we minimize the negative log of $Pr(\mathbf{y}|\mathbf{x})Pr(\mathbf{x})$ instead:

$$-\log Pr(\mathbf{y}|\mathbf{x})Pr(\mathbf{x}) \propto -\log Pr(\mathbf{y}|\mathbf{x}) - \log Pr(\mathbf{x}) \tag{13}$$

The negative log of the likelihood $Pr(\mathbf{y}|\mathbf{x})$ in (11) can be rewritten as:

$$-\log Pr(\mathbf{y}|\mathbf{x}) = k \underbrace{(\log(1-p) - \log(p))}_{\gamma} - N_1 \log(1-p) \tag{14}$$

Because the second term is a constant for fixed N_1 and p, we can ignore it during minimization.

Together with the negative log of the prior $Pr(\mathbf{x})$ in (12), we can write our objective function as follows:

$$\min_{\mathbf{x}} \; \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_0 \, \gamma \; + \; \sigma_0^{-2}\, \mathbf{x}^T\, \mathbf{L}\, \mathbf{x} \; + \; \sigma_1^{-2}\, \mathbf{x}^T\, \mathbf{L}^2\, \mathbf{x} \tag{15}$$

5. ALGORITHM DEVELOPMENT

5.1. Iterative Reweighted Least Square Algorithm

To solve (15), we employ the following optimization strategy. We first replace the l_0-norm in (15) with a weighted l_2-norm:

$$\min_{\mathbf{x}} \; (\mathbf{y} - \mathbf{D}\mathbf{x})^T \mathbf{U} (\mathbf{y} - \mathbf{D}\mathbf{x}) \gamma \; + \; \sigma_0^{-2}\, \mathbf{x}^T\, \mathbf{L}\, \mathbf{x} \; + \; \sigma_1^{-2}\, \mathbf{x}^T\, \mathbf{L}^2\, \mathbf{x} \tag{16}$$

where $\mathbf{U}$ is a $N_1 \times N_1$ diagonal matrix with weights $u_1, \ldots, u_{N_1}$ on its diagonal. In other words, the fidelity term is now a weighted sum of label differences: $(\mathbf{y} - \mathbf{D}\mathbf{x})^T \mathbf{U} (\mathbf{y} - \mathbf{D}\mathbf{x}) = \sum_{i=1}^{N_1} u_i (y_i - x_i)^2$.

The weights u_i should be set so that the weighted l_2-norm mimics the l_0-norm. To accomplish this, we employ the *iterative reweighted least square* (IRLS) strategy [4], which has been proven to have superlinear local convergence, and solve (16) iteratively, where the weights $u_i^{(t+1)}$ of iteration $t+1$ is computed using solution $x_i^{(t)}$ of the previous iteration t, *i.e.*,

$$u_i^{(t+1)} = \frac{1}{(y_i - x_i^{(t)})^2 + \epsilon} \tag{17}$$

for a small $\epsilon > 0$ to maintain numerical stability. Using this weight update, we see that the weighted quadratic term $(\mathbf{y} - \mathbf{D}\mathbf{x})^T \mathbf{U} (\mathbf{y} - \mathbf{D}\mathbf{x})$ mimics the original l_0-norm $\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_0$ in the original objective (15) when the solution $\mathbf{x}$ converges.

5.2. Closed-Form Solution per Iteration

For a given weight matrix $\mathbf{U}$, it is clear that the objective (16) is a unconstrained quadratic programming problem with three quadratic terms. One can thus derive a closed-form solution by taking the derivative with respect to $\mathbf{x}$ and equating it to zero, resulting in:

$$\mathbf{x}^* = \left(\gamma \mathbf{D}^T \mathbf{U} \mathbf{D} + \sigma_0^{-2} \mathbf{L} + \sigma_1^{-2} \mathbf{L}^T \mathbf{L}\right)^{-1} \gamma \mathbf{D}^T \mathbf{U}^T \mathbf{y} \tag{18}$$

5.3. Initialization

It is clear that the IRLS strategy converges to a local minimum in general, and thus it is important to start the algorithm with a good initial solution $\mathbf{x}^{(0)}$. To initialize $x_i^{(0)}$ so that $u_i^{(1)}$ can be computed using (17), we perform the following initialization procedure.

1. Initialize $\mathbf{x}$ by thresholding the solution of (18) with observed $\mathbf{y}$, using the identity matrix $\mathbf{I}$ as the weight matrix $\mathbf{U}$:

$$x_i = \begin{cases} 1, & x_i^* > 0 \\ -1, & x_i^* < 0. \end{cases} \tag{19}$$

2. Identify the entry x_i that minimizes the signal smoothness prior, *i.e.*,

$$i^* = \arg\min_i \; \sigma_0^{-2} \mathbf{x}_{-i}^T \mathbf{L}\, \mathbf{x}_{-i} + \sigma_1^{-2} \mathbf{x}_{-i}^T \mathbf{L}^2\, \mathbf{x}_{-i} \tag{20}$$

where $\mathbf{x}_{-i}$ is the label vector $\mathbf{x}$ with i flipped (*i.e.* convert 1 into -1 or vice versa).

3. If $\mathbf{x}_{-i^*}$ results in a smaller objective function (15), set $\mathbf{x}$ to $\mathbf{x}_{-i^*}$ and goto step 2. Otherwise stop.

The above initialization procedure exploits the fact that the ground truth signal $\mathbf{x}$ contains binary labels, and thus each entry x_i deviates from noise-corrupted y_i by at most 1. In subsequent iterations, computed $\mathbf{x}$ will not be restricted to be a binary vector to reflect confidence level, as discussed earlier.

5.4. Interpreting Computed Solution $\hat{\mathbf{x}}$

After the IRLS algorithm converges to a solution $\hat{\mathbf{x}}$, we interpret the classification results as follows. We perform thresholding by a predefined value τ on $\hat{\mathbf{x}}$ to divide it into three parts, including the rejection option for ambiguous items

$$x_i = \begin{cases} 1, & x_i^* > \tau \\ Rejection, & -\tau < x_i^* < \tau \\ -1, & x_i^* < -\tau. \end{cases} \quad (21)$$

Note that a multiclass classification problem can be reduced to multiple binary classification problems via the one-vs.-rest rule or the one-vs.-one rule [19]. It can then be solved using our proposed graph-based binary classifier successively.

6. EXPERIMENTATION

6.1. Experimental Setup

(a) female 1 (b) female 2 (c) male 1 (d) male 2

Fig. 3. Examples of images in gender classification dataset.

(a) face 1 (b) face 2 (c) non-face 1

Fig. 4. Examples of face and non-face images.

We tested our proposed algorithm against two schemes: i) a more robust version of the famed Adaboost called RobustBoost[3] that claims robustness against label noise, and ii) SVM with a RBF kernel. The first dataset is a gender classification dataset consists of 5300 images of the frontal faces of celebrities from FaceScrub dataset [4], where half of them are male and the other half are female. Example images from the dataset are shown in Fig. 3. We normalize the face images to 400×400 pixels and extracted space LBP features with a cell size of 25×25 pixels. To test the robustness of different classification schemes, we randomly selected a portion of images from the training set and reversed their labels. All the classifiers were then trained using the same set of features and labels. The test set was classified by the classifiers and the results are compared with the ground truth labels. We also tested the same classifiers using the face detection dataset, which consists of 400 face images from ORL face database provided by ATT Cambridge labs and 800 non-face images. We used half of the dataset (200 face images and 400 non-face images) as the training set and the other half as the test set. See Fig. 4 for example images.

[3]http://arxiv.org/pdf/0905.2138.pdf

[4]http://vintage.winklerbros.net/facescrub.html

Table 1. Classification error and rejection rate in gender detection for competing schemes under different training label errors (training set: 1000/1000)

% label noise	5%	10%	15%	25%
Graph ($\sigma_1^{-2} = 0$)	0.07/1.59%	0.31/1.86%	1.42/3.60%	5.39/7.42%
Graph ($\sigma_1^{-2} = 0.5$)	0.00/1.86%	0.23/2.98%	0.80/5.67%	2.47/8.51%
Graph ($\sigma_1^{-2} = 0.9$)	0.00/2.30%	0.12/3.54%	0.49/6.73%	0.67/11.62%
RobustBoost	4.02%	6.06%	9.74%	24.57%
SVM-RBF	4.30%	7.84%	17.32%	40.43%

Table 2. Classification error and rejection rate in face / non-face dataset for competing schemes under different training label errors (training set: 300/300)

% label noise	5%	10%	15%	25%
Graph ($\sigma_1^{-2} = 0$)	0.00/0.48%	0.46/0.59%	0.86/0.87%	1.69/2.63%
Graph ($\sigma_1^{-2} = 0.5$)	0.00/0.82%	0.00/1.03%	0.00/1.85%	0.77/3.34%
Graph ($\sigma_1^{-2} = 0.9$)	0.00/0.91%	0.00/1.32%	0.00/2.41%	0.00/3.93%
RobustBoost	2.23%	3.13%	4.04%	13.76%
SVM-RBF	3.31%	5.39%	7.55%	30.68%

6.2. Experimental Results

The resulting classification error and rejection rate for different classifiers are presented in Table 1, where the percentage of randomly erred training labels ranges from 5% to 25%. In the experiment, we kept σ_0^{-2} constant and varied σ_1^{-2} to induce different rejection rates. We observe that our graph-signal recovery scheme (`graph`) achieved lower classification error when compared to RobustBoost and SVN-RBF at all training label error rates. In particular, at 25% label error rate, our proposal can achieve very low error rates of 5.39%, 2.47% and 0.67% at the cost of rejection rates of 7.42%, 8.51% and 11.62% respectively. In comparison, Robustboost and SVM suffer from severe classification error rate of 24.57% and 40.43% respectively, which is much higher than the sum of error and rejection rate observed in our proposal.

The results also show that by assigning a larger σ_1^{-2}, we can induce a lower classification error rate at the cost of a slightly higher rejection rate. In different applications, a user may define the desired classifier performance as a weighted sum of classification error and rejection rate, as done in [20]. Using our algorithm, a user can thus tune σ_1^{-2} to adjust the preference of classification error versus rejection rate.

The results for face detection dataset are shown in Table 2. We observe similar trends where our proposed algorithm outperforms RobustBoost and SVM-RBF significantly in classification error rate.

7. CONCLUSION

Due to the sheer size of user-generated content in social media, label noise is unavoidable in the training data in a semi-supervised learning scenario. In this paper, we propose a new method for robust graph-based classifier learning via a graph-signal restoration formulation, where the desired signal (label assignments) and its gradient are assumed to be smooth with respect to a properly constructed graph. We describe an iterative reweighted least square (IRLS) algorithm to solve the problem efficiently. Experimental results show that our proposed algorithm outperforms regular SVM and noisy label learning schemes in the literature noticeably.

8. REFERENCES

[1] A. Brew, D. Greene, and P. Cunningham, "The interaction between supervised learning and crowdsourcing," in *Computational Social Science and the Wisdom of Crowds Workshop at NIPS*, Whistler, Canada, December 2010.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in *IEEE Signal Processing Magazine*, May 2013, vol. 30, no.3, pp. 83–98.

[3] K. Bredies and M. Holler, "A TGV-based framework for variational image decompression, zooming and reconstruction. part i: Analytics," in *SIAM Jour*, 2015, vol. 8, no.4, pp. 2814–2850.

[4] I. Daubechies, R. Devore, M. Fornasier, and S. Gunturk, "Iteratively re-weighted least squares minimization for sparse recovery," in *Communications on Pure and Applied Mathematics*, January 2010, vol. 63, no.1, pp. 1–38.

[5] B. Frenay and A. Kaban, "Editorial: Special issue on advances in learning with label noise," in *Elsevier: Neurocomputing*, July 2015, vol. 160, pp. 1–2.

[6] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph," in *Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, July 2011.

[7] Y. Wang and A. Pal, "Detecting emotions in social media: A constrained optimization approach," in *Twenty-Fourh International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, July 2015.

[8] A. Guillory and J. Bilmes, "Label selection on graphs," in *Twenty-Third Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, December 2009.

[9] L. Zhang, C. Cheng, J. Bu, D. Cai, X. He, and T. Huang, "Active learning based on locally linear reconstruction," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 2014, vol. 33, no.10, pp. 2026–2038.

[10] S. Chen, A. Sandryhaila, J. Moura, and J. Kovacevic, "Signal recovery on graphs: Variation minimization," in *IEEE Transactions on Signal Processing*, September 2015, vol. 63, no.17, pp. 4609–4624.

[11] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, August 2014.

[12] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *IEEE International Workshop on Multimedia Signal Processing*, Pula, Italy, October 2013.

[13] J. Pang, G. Cheung, W. Hu, and O. C. Au, "Redefining self-similarity in natural images for denoising using graph signal gradient," in *APSIPA ASC*, Siem Reap, Cambodia, December 2014.

[14] J. Pang, G. Cheung, A. Ortega, and O. C. Au, "Optimal graph Laplacian regularization for natural image denoising," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015.

[15] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Symposium on Graph Signal Processing in IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, TX, December 2013.

[16] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation of graph structured data," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 2013.

[17] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. Au, "Image bit-depth enhancement via maximum-a-posteriori estimation of graph AC component," in *IEEE International Conference on Image Processing*, Paris, France, October 2014.

[18] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Inter-block soft decoding of JPEG images with sparsity and graph-signal smoothness priors," in *IEEE International Conference on Image Processing*, Quebec City, Canada, September 2015.

[19] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2007.

[20] C. Chow, "On optimum recognition error and reject tradeoff," in *IEEE Transactions on Information Theory*, January 1970, vol. 16, pp. 41–46.