Joint Learning of Similarity Graph and Image Classifier from Partial Labels

Yu Mao*, Gene Cheung*, Chia-Wen Lin† and Yusheng Ji*
* National Institute of Informatics, Tokyo, Japan
E-mail: {mao, cheung, kei}@nii.ac.jp Tel/Fax: +81-3-4212-2567
† National Tsing Hua University, Hsinchu, Taiwan
E-mail: cwlin@ee.nthu.edu.tw Tel/Fax: +886-3-573-1152

***Abstract*—Learning of a binary classifier from partial labels is a fundamental and important task in image classification. Leveraging on recent advance in graph signal processing (GSP), a recent work poses classifier learning as a graph-signal restoration problem from partial observations, where the ill-posed problem is regularized using a graph-signal smoothness prior. In this paper, we extend this work by using the same smoothness prior to refine the underlying similarity graph also, so that the same graph-signal projected on the modified graph will be even smoother. Specifically, assuming an edge weight connecting two vertices i and j is computed as the exponential kernel of the weighted sum of feature function differences at the two vertices, we find locally "optimal" feature weights via iterative Newton's method. We show that the conditioning of the Hessian matrix reveals redundancy in the feature functions, which thus can be eliminated for improved computation efficiency. Experimental results show that our joint optimization of the classifier graph-signal and the underlying graph has better classification performance then the previous work and spectral clustering.**

I. Introduction

Semi-supervised learning—the learning of a classifier from partially available labels—is an important task in image classification. Labels are often only partially available in many practical settings such as social media like Facebook and Instagram, where *user-generated content* (UGC) like selfies is growing rapidly, but labeling of this vast content into meaningful categories requires costly human labor.

Leveraging on recent advances in *graph signal processing* (GSP) [1], in one recent work [2] the authors pose binary classifier learning as a graph-signal restoration problem from partial observation, where a signal sample x_i at vertex i takes on binary value $\{-1, 1\}$ to denote event class for this vertex. The ill-posed problem is regularized using a *graph-signal smoothness prior*: an assumption that the desired signal $\mathbf{x}$ is smooth with respect to an underlying similarity graph $\mathcal{G}$ with respective vertex and edge sets $\mathcal{V}$ and $\mathcal{E}$. The graph-signal smoothness prior has been used successfully in many signal restoration problems, such as denoising [3–5], interpolation [6–8], bit-depth enhancement [9] and JPEG de-quantization [10, 11]. Simulation results in [2] for two image datasets with varying amount of label noise show that the constructed graph-based classifier outperforms two competing learning approaches in the literature noticeably.

Extending on [2], in this paper, using the same smoothness prior we refine the underlying graph also, so that the regularization term computed using the same graph-signal $\mathbf{x}$ will result in an even smaller value. The key idea is the following. Partial labels provide crucial information to restore the target graph-signal (classifier), but surely the same information can be used to improve the similarity graph also, if the labels are inconsistent with the graph, and the graph is deemed less trustworthy than the labels. This is often the case in practice, where the labels are collected painstakingly from domain experts, while similarity graphs are constructed in an ad-hoc manner, *e.g.*, edge weights are computed using feature functions selected *a priori* without understanding of their effects on eventual classification performance. Note that this ability to improve similarity graphs using partial labels is not possible in unsupervised learning, where techniques like spectral clustering [12] are tasked to partition graph vertices into two clusters with no label information, and thus must rely solely on the accuracy of the underlying graph.

Specifically, using the same graph-smoothness objective function as [2], we compute locally optimal feature weights iteratively using the Newton's method [13]. We show that the conditioning of the Hessian matrix reveals redundancy in the feature functions, which thus can be eliminated for improved computation efficiency. Experimental results show that out joint optimization of classifier graph-signal and the underlying graph has better classification performance than [2] and spectral clustering.

The outline of the paper is as follows. We review basic GSP concepts in Section II. In Section III, we pose classifier learning as a graph-signal restoration problem, and then formulate the graph learning problem using the same objective. We discuss also how the Newton's method can be used to compute locally optimal weights. Finally, we present experimental results and conclusion in Section IV and V, respectively.

II. Smoothness of Graph-Signals

A. Preliminaries

GSP is the study of signals on structured data kernels described by graphs [1]. 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:

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

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.*,

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

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 [1], *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).

B. Smoothness of Graph-signals

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:

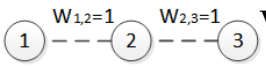
$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} w_{i,j} \left(x_i - x_j\right)^2 \tag{3}$$

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 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 $\mathbf{D}$.

Fig. 1 shows an example of a graph $\mathcal{G}$ with three vertices, and the corresponding weighted adjacency matrix $\mathbf{W}$ and degree matrix $\mathbf{D}$. The combinatorial graph Laplacian $\mathbf{L}$ in this case is:

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

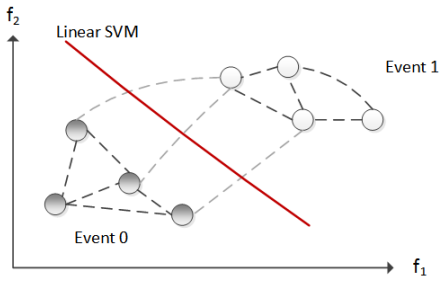


Fig. 2: Example of a constructed graph $\mathcal{G}$ for binary-event classification with two features f_1 and f_2. A linear SVM would dissect the space into two for classification.

III. Problem Formulation

A. Problem Definition

Given partially observed labels $\mathbf{y} \in \{-1, 1\}^M$, we seek to recover a graph-signal $\mathbf{x} \in \{-1, 1\}^N$, $M \ll N$, to correctly label the remaining $N - M$ media events. Denote by $\mathbf{D}$ a $M \times N$ binary matrix, $D_{i,j} \in \{0, 1\}$, that selects M entries from $\mathbf{x}$ that correspond to observed labels in $\mathbf{y}$. Our optimization seeks the *smoothest* signal $\mathbf{x}$ with respect to graph Laplacian $\mathbf{L}$ such that $\mathbf{x}$ agrees with observation $\mathbf{y}$:

$$\min_{\mathbf{x}} \mathbf{x}^T\mathbf{L}\mathbf{x}, \qquad \text{s.t. } \mathbf{D}\mathbf{x} = \mathbf{y} \tag{6}$$

Fig. 2 shows an example of a graph in a 2-dimensional features space. We see that the vertices cluster into two groups corresponding to event 0 and 1 in the feature space, and thus partial labels will propagate correct information to neighboring missing labels via the smoothness prior.

We can equivalently solve the Lagrangian relaxed version of (6) using Lagrange multiplier λ:

$$\min_{\mathbf{x}} \mathbf{x}^T\mathbf{L}\mathbf{x} + \lambda \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2 \tag{7}$$

Variant of this optimization has already been proposed in [2].

The key observation in this work is that *the graph-signal smoothness prior can in turn be used to improve the construction of the similarity graph, resulting in an even smaller smoothness objective.* Assume that edge weight $w_{i,j}$ connecting vertices i and j is computed using K pre-selected *feature functions* $f_k(\,)$ evaluated at i and j and a Gaussian kernel, as done in [3–11]:

$$w_{i,j} = \exp\left\{-\sum_{k=1}^{K} c_k \left(f_k(i) - f_k(j)\right)^2\right\} \tag{8}$$

where c_k are the *feature weights* that determine the relative importance of the K feature functions. We can now optimize $\mathbf{c} = [c_1, \ldots, c_K]^T$ formally using the same smoothness prior:

$$\min_{\mathbf{c}} \mathbf{x}^T\mathbf{L}_{\mathbf{c}}\mathbf{x}, \qquad \text{s.t. } \mathbf{c}^T\mathbf{1} \leq C \tag{9}$$

where constraint $\mathbf{c}^T\mathbf{1} \leq C$ is necessary to prevent the trivial solution when $c_k = \infty$ and $\mathbf{x}^T\mathbf{L}\mathbf{x} = 0$. Note also that graph Laplacian $\mathbf{L}_{\mathbf{c}}$ is implicitly a function of $\mathbf{c}$ that determines edge weights $w_{i,j}$, hence the subscript.

We write the Lagrangian relaxed version as follows:

$$\min_{\mathbf{c}} g(\mathbf{c}) = \mathbf{x}^T \mathbf{L}_{\mathbf{c}} \mathbf{x} + \mu\, \mathbf{c}^T \mathbf{1} \tag{10}$$

where $\mu > 0$ is selected large enough so that $\mathbf{c}^T \mathbf{1} \approx C$.

B. Newton's Descent Method

There are no closed-form solutions for (10). To solve it, we can employ the Newton's method [13] to iteratively converge to a locally optimal solution:

$$\mathbf{c}^{t+1} = \mathbf{c}^t - \left(\nabla^2 g(\mathbf{c}^t)\right)^{-1} \nabla g(\mathbf{c}^t) \tag{11}$$

where $\mathbf{c}^t$ is the solution at iteration t, $\nabla g(\mathbf{c}^t)$ and $\nabla^2 g(\mathbf{c}^t)$ are respectively the gradient vector and Hessian matrix of $g(\mathbf{c})$ evaluated at $\mathbf{c}^t$. Clearly (11) is computationally feasible only if Hessian $\nabla^2 g(\mathbf{c}^t)$ is invertible, *i.e.*, it does not contain eigenvalue 0. We will address this issue shortly.

To simplify notation, we define the following terms:

$$\phi_k(i,j) = \left(f_k(i) - f_k(j)\right)^2 \tag{12}$$

$$\delta(i,j) = \left(x_i - x_j\right)^2 \tag{13}$$

The p-th entry in the gradient vector $\nabla g(\mathbf{c})$ is:

$$\frac{d\, g(\mathbf{c})}{d\, c_p} = \sum_{(i,j)\in\mathcal{E}} e^{-\sum_{k=1}^{K} c_k \phi_k(i,j)} (-1)\, \phi_p(i,j)\, \delta(i,j) + \mu \tag{14}$$

For the Hessian matrix $\nabla^2 g(\mathbf{c}^t)$, the p-th diagonal entry is:

$$\frac{d^2\, g(\mathbf{c})}{d\, c_p^2} = \sum_{(i,j)\in\mathcal{E}} e^{-\sum_{k=1}^{K} c_k \phi_k(i,j)} \phi_p^2(i,j)\, \delta(i,j) \tag{15}$$

The (p,q)-th off-diagonal entry in $\nabla^2 g(\mathbf{c}^t)$ takes a similar form:

$$\frac{d^2\, g(\mathbf{c})}{d\, c_p\, c_q} = \sum_{(i,j)\in\mathcal{E}} e^{-\sum_{k=1}^{K} c_k \phi_k(i,j)} \phi_p(i,j)\, \phi_q(i,j)\, \delta(i,j) \tag{16}$$

From (16) we see that the (p,q)-th entry is the same as the (q,p) entry, and so Hessian matrix $\nabla^2 g(\mathbf{c}^t)$ is symmetric. More importantly, we show next that $\nabla^2 g(\mathbf{c}^t)$ is invertible only if the feature functions $f_k(i)$ are linearly independent, so that the Newton's method (11) can be used to solve (10).

Lemma 3.1: $\nabla^2 g(\mathbf{c}^t)$ has eigenvalue 0 if feature functions $f_k(i)$ are linearly dependent.

Proof: We will prove by contradiction. Without loss of generality, suppose a feature $f_{p+1}(i)$ is a linear combination of p previous features; *i.e.,* $\phi_{p+1}(i,j) = \sum_{k=1}^{p} a_k \phi_k(i,j)$. Suppose Hessian $\nabla^2 g(\mathbf{c}^t)$ does not contain eigenvalue 0. The $p+1$-th row of Hessian $\nabla^2 g(\mathbf{c}^t)$ has off-diagonal entry $(p+1, q)$ equals to

$$\sum_{(i,j)\in\mathcal{E}} e^{-\sum_{k=1}^{K} c_k \phi_k(i,j)}\, \delta(i,j)\, \phi_q(i,j) \sum_{k=1}^{p} a_k \phi_k(i,j) \tag{17}$$

The diagonal term $(p+1, p+1)$ of Hessian $\nabla^2 g(\mathbf{c}^t)$ will be a special case of (17), where $\phi_q(i,j) = \phi_{p+1}(i,j) = \sum_{k=1}^{p} a_k \phi_k(i,j)$.

One can now verify that the $p+1$-th row of Hessian $\nabla^2 g(\mathbf{c}^t)$ is exactly the same as a linear combination of the first p rows using weight a_k for the k-th row. A matrix with linearly dependent rows implies that the matrix has eigenvalue 0. A contradiction. ■

The lemma means that the Hessian $\nabla^2 g(\mathbf{c}^t)$ is invertible if each feature function $f_k(i)$ is *innovative*, *i.e.*, it provides new information for classification and thus is not a simple linear combination of other features. This tends to be the case when features are selected carefully [14].

The corollary of the lemma is that the conditioning of Hessian $\nabla^2 g(\mathbf{c}^t)$ is actually informative in checking if current feature functions $f_k(i)$'s are redundant and can be removed. In fact, an eigenvalue close to 0 would imply that there exists one or more feature functions that are minimally useful in providing new information for classification. Given that the cost of computing a feature function for all vertices in a large graph can be significant, using the smallest innovative set of feature functions can reduce the computation cost.

After a new set of feature weighs $\mathbf{c}^{t+1}$ has been computed using (11), the optimal graph-signal $\mathbf{x}$ given the graph can be solved again via (7). The procedure repeats until both the signal $\mathbf{x}$ and the feature weights $\mathbf{c}$ converge.

IV. Experimentation

A. Experiment Setup

We tested our proposal against two schemes: i) the classification by thresholding the result of (7) without optimizing the graph as done in [2], and ii) spectrum clustering [12] using the optimized graph. As spectrum clustering only clusters the samples into groups, we manually attempted the two possible ways to label the groups, and chose the one with the lower error rate. The experiment was conducted on two different data sets. The first one was `skin segmentation` data set[1], which uses the RGB values of colors as features and labels the colors indicating whether they are possible for human skin. By using the above information, we can achieve a fast preliminary skin segmentation for an image. Example of this fast skin segmentation method from the dataset [15] is shown in Fig. 3. A subset of 2000 samples was used in our experiment, and each sample contained the RGB value as three features.

(a)Input image (b)Result image

Fig. 3: Example of skin segmentation by color, the white area in the result image corresponds to possible skin area.

The second dataset was `pima indians diabetes` dataset[2], which predicts health condition: whether the individual is a diabetes patient, based on her health indicators.

[1]https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation

[2]https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

TABLE I: Classification accuracy in Skin Segmentation data set

training set size	600	300
proposed method	93.06%	89.24%
without optimizing graph	91.23%	85.67%
spectrum clustering	60.23%	62.56%

TABLE II: Classification accuracy in Pima Indians Diabetes data set

training set size	200	100
proposed method	71.81%	68.13%
without optimizing graph	68.78%	65.34%
spectrum clustering	65.31%	63.95%

The dataset contains 768 samples, each corresponds to a Pima Indian female, and eight of their health indicators, such as diastolic blood pressure, triceps skin fold thickness and body mass index are recorded as features. For both datasets, we randomly selected a subset as training set and interpolated the rest for validation.

B. Experimental Results

The experiment results for the `skin segmentation` dataset are shown in Table I. The results show that the optimization of the graph improves the classification results by 1.83% when we have 600 samples in the training set and 3.57% when only 300 samples are used for training. Second, we observe the limitation of spectrum clustering, which does not utilize observed labels, when applied to classification problems. Similar trends are observed from the results for `pima indian diabetes` dataset, shown in Table II.

We also tested the convergence speed of our proposed Newton-method-based graph optimization against gradient descent optimization [13]. The iteration numbers needed to reach convergence are listed in Table. III. We observe that Newton's method can reach convergence faster than gradient descent in our application. During the experimentation, we did not observe any Hessian matrix with eigenvalue 0 or close to 0, which means that the feature functions were all innovative and the Hessian invertible.

V. Conclusion

Learning of a graph-based binary classifier from a similarity graph and partial labels has been studied in a recent work [2] that formulates a graph-signal restoration problem, regularized using a graph-signal smoothness prior. In this paper, we use the same graph-signal smoothness prior to improve construction of the similarity graph as well, so that the graph-signal regularization term computed on the same graph-signal will result in an even smaller value. Assuming that an edge weight $w_{i,j}$ is computed using a Gaussian kernel of the linear combination of feature function differences evaluated at vertices i and j, we propose to compute locally optimal feature weights via Newton's method. We show that the Hessian matrix is not invertible if the feature functions are linearly dependent; the corollary is that the conditioning of the Hessian can be used as a criteria to eliminate redundant or minimally useful feature functions to reduce complexity. Experimental results show that our joint optimization of the graph-based classifier and the similarity graph can lead to better classification performance compared to [2] and spectral clustering.

TABLE III: the number of iterations of Newton's method and gradient descent in graph optimization

	Newton's Method	Gradient Descent
Skin/600	5	8
Skin/300	6	10
Pima/200	6	8
Pima/100	8	13

References

[1] 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.

[2] Y. Mao, G. Cheung, C.-W. Lin, and Y. Ji, "Image classifier learning from noisy labels via generalized graph smoothness priors," in *IEEE IVMSP Workshop*, Bordeaux, France, July 2016.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Y. Mao, G. Cheung, and Y. Ji, "Image interpolation during DIBR view synthesis using graph Fourier transform," in *3DTV-Conference*, Budapest, Hungary, July 2014.

[9] 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.

[10] 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.

[11] W. Hu, G. Cheung, and M. Kazui, "Graph-based dequantization of block-compressed piecewise smooth images," in *IEEE Signal Processing Letters*, February 2016, vol. 23, no.2, pp. 242–246.

[12] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[13] Dimitri P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[14] I. Guyon et al., "Gene selection for cancer classification using support vector machines," in *Machine Learning*, 2002, vol. 46, (1-3), pp. 389–422.

[15] R. B. Bhatt, G. Sharma, A. Dhall, and S. Chaudhury, "Efficient skin region segmentation using low complexity fuzzy decision tree model," in *IEEE INDICON 2009*, Ahmedabad, India, December 2009.