Supplemental Material for "Wasserstein Soft Label Propagation on Hypergraphs: Algorithm and Generalization Error Bounds"

Tingran Gao, Shahab Asoodeh, Yi Huang, and James Evans
The University of Chicago
{trg17, shahab, yhuang10, jevans}@uchicago.edu

Abstract

Below are the supplemental material for the paper "Wasserstein Soft Label Propagation on Hypergraphs: Algorithm and Generalization Error Bounds" submitted to AAAI 2019

Proof of Lemma 1

The conditions on Φ can be written as

$$\left[\frac{t_i}{m\gamma}+\deg\left(i\right)\right]\Phi\left(i\right)-\sum_{j:j\sim i}\Phi\left(j\right)\geq 0 \qquad 1\leq i\leq \ell \tag{1}$$

$$\deg\left(i\right)\Phi\left(i\right)-\sum_{j:j\sim i}\Phi\left(j\right)=0 \qquad \ell+1\leq i\leq n \tag{2}$$

where $\deg\left(i\right)\geq 1$ is the degree of vertex i in graph G. First, we assert that the minimum of Φ must be attained among the vertices $1,\cdots,\ell$, for otherwise, if $\ell+1\leq i_*=\arg\min_{i\in V}\Phi\left(i\right)\leq n$, then by (2) we have

$$\begin{aligned}\deg\left(i_*\right)\Phi\left(i_*\right)&=\sum_{j:j\sim i_*}\Phi\left(j\right)\\&\geq\sum_{j:j\sim i_*}\Phi\left(i_*\right)=\deg\left(i_*\right)\Phi\left(i_*\right)\end{aligned}$$

which implies $\Phi\left(j\right)=\Phi\left(i_*\right)$ for all vertices $j\sim i_*$. This argument can be repeated until the constant value propagates into the vertices within $1,\cdots,\ell$, and the assertion follows from the connectivity of the graph. The assertion for the maximum can be established analogously. Next we argue that the minimum of Φ on the vertices of G must be non-negative. Assume the contracy, i.e. the minimum attained at $i_*\in[1,\ell]$ is strictly negative, then by (1) we have

$$\begin{aligned}0&\leq\left[\frac{t_{i_*}}{m\gamma}+\deg\left(i_*\right)\right]\Phi\left(i_*\right)-\sum_{j:j\sim i_*}\Phi\left(j\right)\\&=\frac{t_{i_*}}{m\gamma}\Phi\left(i_*\right)+\sum_{j:j\sim i_*}\left[\Phi\left(i_*\right)-\Phi\left(j\right)\right]<0\end{aligned}$$

where the strict inequalty follows from the counter-assumption $\Phi\left(i_*\right)<0$. This contradiction completes our proof that $\Phi\geq 0$ on the entire graph G.

Proof of Theorem 4

Following the same argument as in the proof of (Belkin, Matveeva, and Niyogi 2004, Theorem 5), we can assume without loss of generality that S, S' differ by a new point $(v_m,\mu_m)\leftrightarrow(v'_m,\mu'_m)$; the other case where only the multiplicities differ can be treated similarly. By our assumption (22), the two averages differ by at most an amount of

$$|\bar{y}_s-\bar{y}'_s|\leq\frac{2M_s}{m}.$$

For simplicity, introduce temporary notations

$$A:=T_\ell+m\gamma L,\qquad B:=T'_\ell+m\gamma L.$$

Using the simple fact that the 2-norm dominate the ∞-norm, we have

$$\begin{aligned}&\|\Phi_s^*-\Phi_s'^*\|_\infty\leq\|\Phi_s^*-\Phi_s'^*\|_2\\&\leq\frac{2M_s}{m}+\left\|A^{-1}\left(\mathbf{y}_s-\bar{y}_sT_\ell\mathbf{1}\right)-B^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)\right\|_2\\&\leq\frac{2M_s}{m}+\left\|A^{-1}\left(\mathbf{y}_s-\bar{y}_sT_\ell\mathbf{1}\right)-A^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)\right\|_2\\&\qquad+\left\|A^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)-B^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)\right\|_2.\end{aligned}$$

Standard functional analysis argument (the same perturbation reasoning we gave in (18)) tells us that $\left\|A^{-1}\right\|_2\leq\left(m\gamma\lambda_1-T\right)^{-1}$. Together with the observation that

$$\begin{aligned}&\|(\mathbf{y}_s-\bar{y}_sT_\ell\mathbf{1})-(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1})\|_2\\&\leq\|\mathbf{y}_s-\mathbf{y}'_s\|_2+\|\bar{y}_sT_\ell\mathbf{1}-\bar{y}'_sT'_\ell\mathbf{1}\|_2\\&\leq 2M_s+\frac{2M_s}{m}<4M_s\end{aligned}$$

we have

$$\left\|A^{-1}\left(\mathbf{y}_s-\bar{y}_sT_\ell\mathbf{1}\right)-A^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)\right\|_2\leq\frac{4M_s}{m\gamma\lambda_1-T}.$$

In the meanwhile, noting that we also have $\left\|B^{-1}\right\|_2\leq\left(m\gamma\lambda_1-T\right)^{-1}$, and $\|A-B\|_2=\|T'_\ell-T_\ell\|_2\leq\sqrt{2}<3/2$, we conclude that

$$\begin{aligned}&\left\|A^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)-B^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)\right\|_2\\&=\left\|B^{-1}\left(B-A\right)A^{-1}\left(\mathbf{y}'_s-\bar{y}'_sT'_\ell\mathbf{1}\right)\right\|_2\leq\frac{3M_s\sqrt{Tm}}{\left(m\gamma\lambda_1-T\right)^2}.\end{aligned}$$

Putting everything together completes the proof.

Proof of Lemma 2

By the equivalence between (4) and (14), it suffices to show the following fact: for each fixed $s \in [0,1]$, if $\max\left\{\left|F_{\mu_i}^{-1}(s)\right|,\ i=1,\cdots,m\right\} \leq \phi(s)$ then $\|\Phi_s^*\|_\infty \leq \phi(s)$, where Φ_s^* is defined in (20). But this follows straightforwardly from the maximum principle.

Numerical Experiments

Label Propagation Algorithm

Alg. 1 details the label propagation algorithm we use to obtain the results in the next two sections.

The functions Barycenter and WassDist can be any algorithms that calculate the weighted Wasserstein barycenter of a vector of labels L with weights W, and the Wasserstein distance between two input labels, respectively. Note that we introduce another parameter $\alpha > 1$ to adjust the weights of vertices with known labels (in line 5) in order to increase their influences to hyperedge barycenters. Similar techniques are explored in (Shi, Osher, and Zhu 2017; 2018).

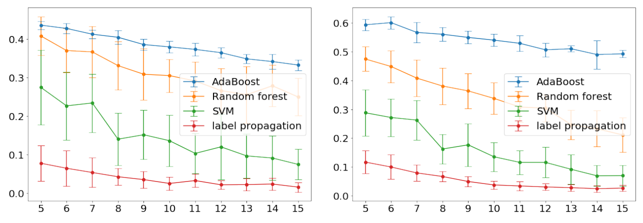
The algorithm relies on the alternating technique in minimizing (9) in each iteration. This technique consists of two steps: (i) first calculates the barycenters $\mathsf{bar}(E)$ of all hyperedges E using the current labels of vertices they contain and treats the derived barycenters as the labels of the hyperedges (lines 21 to 24), and (ii) then calculates the barycenters, i.e. the new labels, of all vertices using labels of the hyperedges incident to them, together with their targeted labels if the latter are known (line 25 to 37). Due to the alternating nature of the algorithm, we call it *alternating label propagation*.

Stochastic Block Model

In the first two experiments, we run label propagation on 3-uniform hypergraphs generated using the stochastic block model (SBM) over 100 vertices that are grouped into either 2 or 3 blocks. More specifically, the probability that a hyperedge $\{v_i, v_j, v_r\}$ exists is $p = 0.01$ if all v_i, v_j, and v_r belong to the same block and is $q = 0.002$ otherwise.

We set the soft labels to be b-dimensional Gaussian distributions, where b is the number of blocks. For any vertex from block i, $i = 1, \ldots, b$, whose label is known, we set the mean of its label to be e_i, where e_i is the base vector with the i-th coordinate being 1 and the rest being 0. The covariance matrix of each known label is set to be $0.05 I_b$, where I_b is the b-dimensional identity matrix. The predicted block assignment of a vertex is the $\arg\max$ of its predicted mean. In both of the experiments, we use $\alpha = 20$ and $\gamma = 10$. We run the experiments with 5 to 15 vertices of known block assignment from each block, and the error bars are obtained by averaging over 20 random selections of vertices with known labels.

We compare the performance of our label propagation approach with with AdaBoost, random forest, and SVM in Fig. 1. We use incidence matrix as the feature matrix in AdaBoost, Random forest, and SVM to solve the classification problem.



(a) SBM with 2 blocks (b) SBM with 3 block

Fig. 1. Comparison of traditional classification algorithms with hypergraph label propagation on SBM.

SBM with two blocks: The hypergraph generated for this experiment has two blocks of sizes 50 and 50, and 629 hyperedges with 388 of them containing vertices from one block.

SBM with three blocks: The hypergraph generated for this experiment has three blocks of sizes 33, 33, and 34, and 384 hyperedges with 182 of them has vertices from one block.

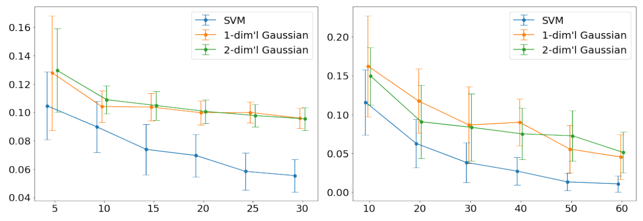
UCI datasets

In the next two experiments, we apply our label propagation as a classification algorithm to the following two datasets with categorical features from the UCI machine learning repository:

Congressional Voting Records: This dataset contains voting records on 16 issues of the 2nd session of the 98th Congress. We form a pair of hyperedges for each issue each of which contains voters who voted "Yay" and "Nay", respectively. For voters whose votes were missing, we don't include them in any of the hyperedges constructed for the corresponding issue. This resilience to the missing data samples illustrates another advantage of applying hypergraph label propagation to classification problems. We test label propagation algorithm with 5, 10, 15, 20, 25, and 30 congressmen and women from each party whose affiliation are given.

Mushrooms: This dataset contains 22 features (e.g., shapes, colors, and habitats, etc) of 8124 mushrooms. We form 97 hyperedges each of which contains mushrooms sharing identical features. We choose 1000 edible and 1000 poisonous mushrooms to run the experiment. We run the algorithm in 6 cases where 10, 20, 30, 40, 50, and 60 mushrooms are given labels from each category.

In both datasets, the soft labels are either 1-dimensional Gaussian distributions $N(+1, 0.01)$ and $N(-1, 0.01)$ or 2-dimensional Gaussian distributions $N((1,0), 0.01 I_2)$ and $N((0,1), 0.01 I_2)$ depending on which class the labelled sample belongs to. The predicted class of a vertex is obtained as follows: For the 1-dimensional case, it is the sign of the mean of its label and for the 2-dimensional case, it is $+1$ if the first coordinate of the mean vector of its label is larger than the second coordinate and -1 otherwise. For both experiments, we set $\alpha = 10$ and $\gamma = 1$. The error bars are obtained by averaging 20 random selections of vertices with known labels. We compare the performance of hypergraph label propagation (as a classification algorithm) with SVM in Fig. 2.



(a) Congressional voting records (b) Mushrooms

Fig. 2. Comparison of SVM with hypergraph label propagation as a classification algorithm.

Discussion of numerical experiments

The above experiments demonstrate that the hypergraph label propagation can serve as a powerful alternative classification algorithm especially when the dataset is structured as a network (for example as in SBM). The reason as to why the traditional classification algorithms may fail on network-like datasets (as illustrated in Fig. 1) is because for these datasets almost all coordinates of a feature vector tend to be identical except for few of them. We can understand these features as describing only local properties of the dataset. Therefore, they can give rise to global characterizations of the datasets, in a substantial way, only when properly “patched” together. Label propagation algorithm provides a novel way of combining features which is shown in Fig. 1 to outperform the classical algorithms.

References

Belkin, M.; Matveeva, I.; and Niyogi, P. 2004. Regularization and Semi-Supervised Learning on Large Graphs. In *International Conference on Computational Learning Theory*, 624–638. Springer.

Shi, Z.; Osher, S.; and Zhu, W. 2017. Weighted nonlocal laplacian on interpolation from sparse data. *Journal of Scientific Computing* 73(2-3):1164–1177.

Shi, Z.; Osher, S.; and Zhu, W. 2018. Generalization of the weighted nonlocal laplacian in low dimensional manifold model. *Journal of Scientific Computing* 75(2):638–656.

Algorithm 1: Alternating label propagation algorithm

```
Data: hypergraph $H = (V, \mathcal{E})$; a subset of vertices $V_0$
      with known labels $\bar{l}(v)$, $\forall v \in V_0$; parameters
      $\alpha, \gamma > 0$, a condition EC for exiting the main
      loop on line 19.
Result: labels $l(v)$, $\forall v \in V$.
1  Randomly initialize labels $l(v)$, $\forall v \in V$
2  for every $E \in \mathcal{E}$ do
3      for every $v \in E$ do
4          if $v \in V_0$ then
5              $W_E(v) = \alpha$
6          else
7              $W_E(v) = 1$
8          end
9      end
10 end
11 for every $v \in V$ do
12     for every $E \in \mathcal{E}$ incident to $v$ do
13         $w_v(E) = 1/|E|$
14     end
15     if every $v \in V_0$ then
16         append vector $W_v$ with $\gamma$
17     end
18 end
19 while EC is not met do
20     Initialize $loss = 0$
21     for every $E \in \mathcal{E}$ do
22         $L_E = (l(v))_{v \in E}$
23         $l(E) = \text{Barycenter}\,(W_E, L_E)$
24     end
25     for every $v \in V$ do
26         $L_v = (l(E))_{E \text{ incident to } v}$
27         if $v \in V_0$ then
28             append $L_v$ with $\bar{l}(v)$
29         end
30         $l(v) = \text{Barycenter}\,(W_v, L_v)$
31         for every $E \in \mathcal{E}$ incident to $v$ do
32             $loss = loss + W_v(E) \cdot \text{WassDist}(l(v), l(E))$
33         end
34         if $v \in V_0$ then
35             $loss = loss + \gamma \cdot \text{WassDist}\,\big(l(v), \bar{l}(v)\big)$
36         end
37     end
38 end
```