Joint Learning of the Graph and the Data Representation for Graph-Based Semi-Supervised Learning

Mariana Vargas, Aurélien Bellet, Pascal Denis
INRIA

Motivation

- Lack of annotated data is a bottleneck in many NLP applications.
- Semi-supervised learning (SSL) algorithms can address this problem by leveraging on unlabeled data.
- But:
  (i) Graph based SSL rely on a *a-priori* graph which is not given in general,
  (ii) and graph learning methods fail to incorporate label information.
- **Our goal is to construct a graph that adapts to the data and the task by jointly learning a graph and a data representation.**
  - The graph provides a smooth topology with respect to the new representation, addressing problem (i),
  - The data representation injects label information in the graph, addressing problem (ii).

Notation and problem setting

- We consider a dataset $L \cup U$ such that $L = \{(x_i, y_i)\}_{i=1}^{l}$ and $U = \{x_i\}_{i=l+1}^{l+u}$.
- Data points x_i lie in some space $\mathcal{X}$ and are considered signals in nodes v_i.
- Labels $y_i \in \{1, \dots, C\}$ are discrete.
- Let $X \in \mathbb{R}^{n \times d}$, $n = l + u$ be the design matrix.
- Let $G = (V, W)$ be a graph with nodes $V = \{v_1, \dots, v_n\}$ and a symmetric nonnegative weighted adjacency matrix $W \in \mathcal{W}$ where $\mathcal{W} = \{W : W \geq 0, \mathrm{diag}(W) = 0, W^\top = W, W \in \mathbb{R}^{n \times n}\}$.
- **We want to find the labeling** $y_{l+1}, \dots, y_{l+u}$.

Model

Formulation

- We propose to learn a **weighted adjacency matrix** W^* and a **representation function** ϕ_{Θ^*} by minimizing a joint objective function f involving labeled and unlabeled data.
- f has the form $f(W, \Theta) = f_1(\Theta) + \alpha[f_2(W) + f_3(W, \Theta)]$ such that
  - f_1 is the representation-specific term of the form

$$f_1(\Theta) = \sum_{\substack{x_i, x_j, x_k \in L \\ y_i = y_j, y_i \neq y_k}} \left[(Z_\Theta)_{ij} - (Z_\Theta)_{ik} + 1\right]_+, \tag{1}$$

    where $(Z_\Theta)_{ij} = ||\phi_\Theta(x_i) - \phi_\Theta(x_j)||$,
  - f_2 is the graph-specific term of the form

$$f_2(W) = \beta ||W||_F^2 - \mathbf{1}^\top \log(\mathbf{1}^\top W), \tag{2}$$

    where β controls the sparsity on the graph,
  - and f_3 is the joint term

$$f_3(W, \Theta) = \mathrm{tr}(W Z_\Theta) = \textstyle\sum_{i,j} W_{ij} (Z_\Theta)_{ij}. \tag{3}$$

- Once obtained we can plug W and $\phi_{\Theta^*}(X)$ in a semi-supervised learning algorithm.

Optimization

- We propose to optimize the cost function $f(W, \Theta)$ by alternating minimization over W and Θ.
- One step **learns a smooth graph** W **with respect to the current representation**,
- The other step **learns the parameters** Θ **of the representation regularized by a smoothness term involving** W.
- We propose to initialize the graph weights to zero and to start by optimizing Θ so that the initial representation focuses only on the (scarce) labeled data.
- We optimize both subproblems, fitting W and fitting Θ, by gradient descent.

Choices of Representation Function

- An option is to choose ϕ_Θ to be a linear mapping $\phi_\Theta(x) = \Theta x$ which transforms the initial d-dimensional representation (typically a word embedding) into a k-dimensional one, with $\Theta \in \mathbb{R}^{k \times d}$ and $k \leq d$.
- Recent work in learning deep contextualized word representations such as ELMo allows to learn a task-specific combination of the token representations obtained at the K layers of the model. In this case, we could choose $\phi_\Theta(x)$ to be a weighted combination of the layer representations of the word x, that is, $\phi_\Theta(x) = \Theta x \in \mathbb{R}^d$ where $\Theta \in \mathbb{R}^K$ is a K-dimensional parameter vector.

Experiments

Synthetic data

- We assume ϕ_Θ to be a linear transformation $\phi_\Theta(X) = \Theta X$.
- We generated a 3-dimensional dataset consisting of 100 points evenly distributed in two classes.
- We have two clusters per class placed far from each other while keeping clusters from different classes closer.
- We randomly picked 60% of the points and removed their labels.
- We compare classification error in LS when using our learned graph.

	Label Spreading
radial	0.493
knn	0.187
ours-fixed-repr	0.227
ours-1-alt	0.120
ours-full-alt	**0.04**

Table: Classification errors on the synthetic dataset.

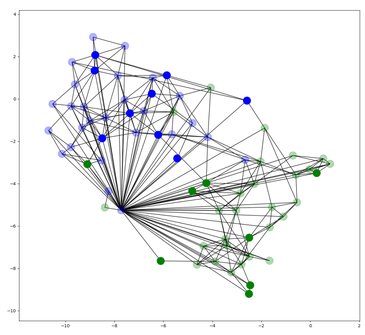

Figure: **fixed-repr** on original data.

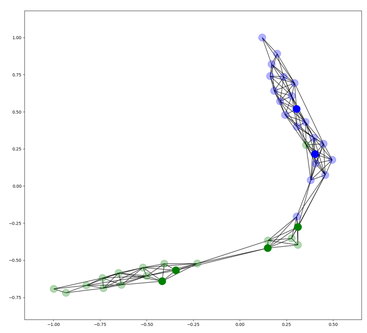

Figure: **ours** after 2nd iteration.

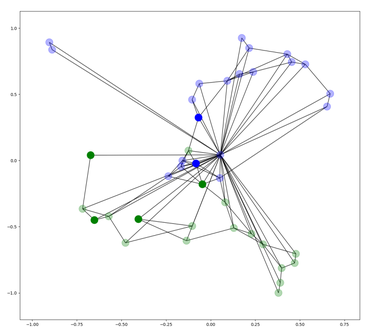

Figure: **ours** after 1st iteration.

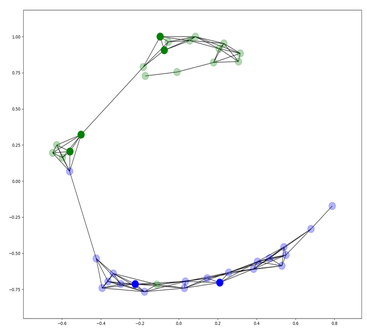

Figure: **ours** after last iteration.

Real data

- Still assuming Φ_Θ to be a linear transformation.
- We now evaluate our method on four classification datasets:
  - **computers** (from 20 News Groups) with classes IBM and Mac ($n = 1945$ documents),
  - **religion** (from 20 News Groups) with classes atheism and Christian ($n = 1796$),
  - **sports** (from 20 News Groups) with classes baseball and hockey ($n = 1993$).
  - **trec**, a question classification data set with $n = 5452$ points in 6 classes: abbreviation, description, entity, human, location, and number.

2*dataset-%	**Baselines**		**Variants of our approach**		
	radial	knn	fixed-repr	1-alt	full-alt
comp-10	.393	.389	.377	**.336**	**.336**
rel-10	.193	.253	.194	.180	**.169**
sports-10	.148	.200	.143	.509	**.065**
trec-10	.323	.394	**.318**	.345	.345
comp-25	.321	.346	.330	**.262**	**.262**
rel-25	.176	.157	.176	**.146**	**.146**
sports-25	.066	.095	.071	**.047**	**.047**
trec-25	.303	.355	**.281**	.289	.289
comp-40	.278	.322	.294	.509	**.233**
rel-40	.153	.166	.144	**.132**	**.132**
sports-40	.063	.098	.063	**.025**	**.025**
trec-40	.263	.300	**.249**	.259	.259
comp-60	.261	.303	.232	**.175**	**.175**
rel-60	.160	.142	.125	**.103**	**.103**
sports-60	.042	.062	.058	**.026**	**.026**
trec-60	.243	.273	.220	.247	.247

Table: Classification error of LS for different graph construction methods and proportions of labeled data.

- The results show that learning the representation along with the graph makes a clear difference.
- Alternating between learning the representation and learning the graph leads to significant gains results in several cases where learning the representation based on the labeled data only leads to severe overfitting.

Ongoing and Future Work

- More expressive graph-based SSL models such as GCN.
- An end-to-end semi-supervised algorithm.
- Improve scalability on large datasets.