Revisiting Semi-Supervised Learning with Graph Embeddings

基于图嵌入的半监督学习

Zhilin Yang ZHILINY@CS.CMU.EDU
William W. Cohen WCOHEN@CS.CMU.EDU
Ruslan Salakhutdinov RSALAKHU@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University

Main Work

We present a semi-supervised learning framework based on graph embeddings. Given a graph between instances, we train an embedding for each instance to jointly predict the class label and the neighborhood context in the graph.

Main Work

We present a semi-supervised learning framework based on graph embeddings. Given a graph between instances, we train an embedding for each instance to jointly predict the class label and the neighborhood context in the graph.

The main highlight of our work is to incorporate embedding techniques into the graph-based semi-supervised learning setting.

What Is Semi-Supervised Learning?

Let L and U be the number of labeled and unlabeled instances. Let $\mathbf{x}_{1:L}$ and $\mathbf{x}_{L+1:L+U}$ denote the feature vectors of labeled and unlabeled instances respectively. The labels $y_{1:L}$ are also given.

What Is Semi-Supervised Learning?

Let L and U be the number of labeled and unlabeled instances. Let $\mathbf{x}_{1:L}$ and $\mathbf{x}_{L+1:L+U}$ denote the feature vectors of labeled and unlabeled instances respectively. The labels $y_{1:L}$ are also given.

Based on both labeled and unlabeled instances, the problem of semi-supervised learning is defined as learning a classifier $f : \mathbf{x} \to y$.

What Is Semi-Supervised Learning?

Let L and U be the number of labeled and unlabeled instances. Let $\mathbf{x}_{1:L}$ and $\mathbf{x}_{L+1:L+U}$ denote the feature vectors of labeled and unlabeled instances respectively. The labels $y_{1:L}$ are also given.

Based on both labeled and unlabeled instances, the problem of semi-supervised learning is defined as learning a classifier $f : \mathbf{x} \to y$.

There are two learning paradigms, transductive learning and inductive learning.

Transductive Learning **VS** Inductive Learning

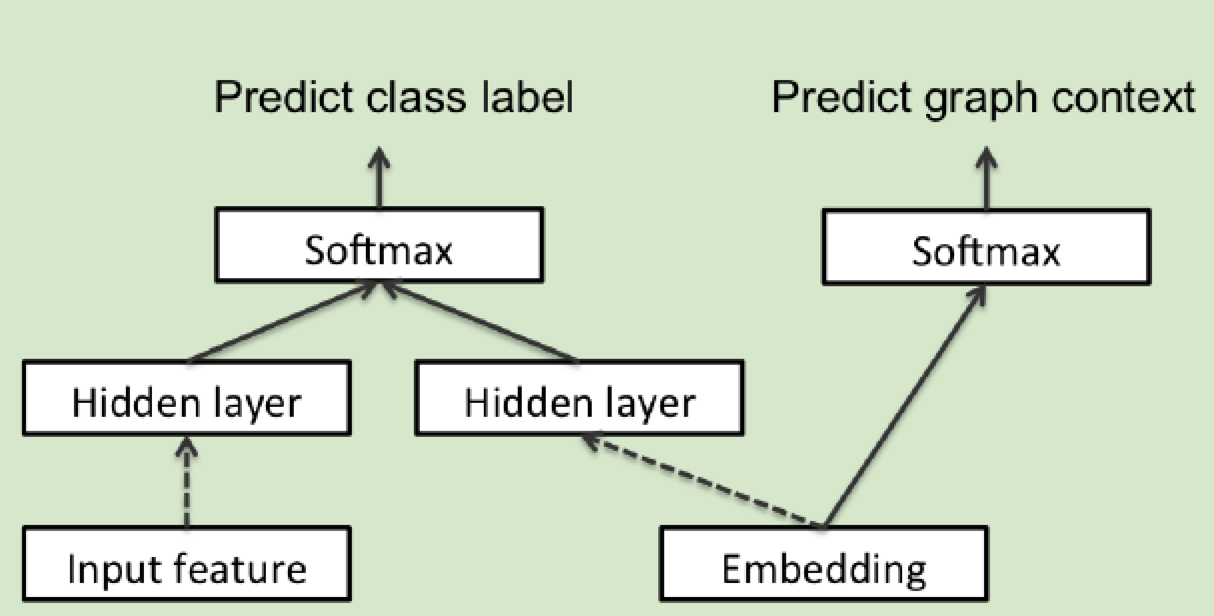
Transductive learning only aims to apply the classifier f on the unlabeled instances observed at training time, and the classifier does not generalize to unobserved instances.

Transductive Learning VS Inductive Learning

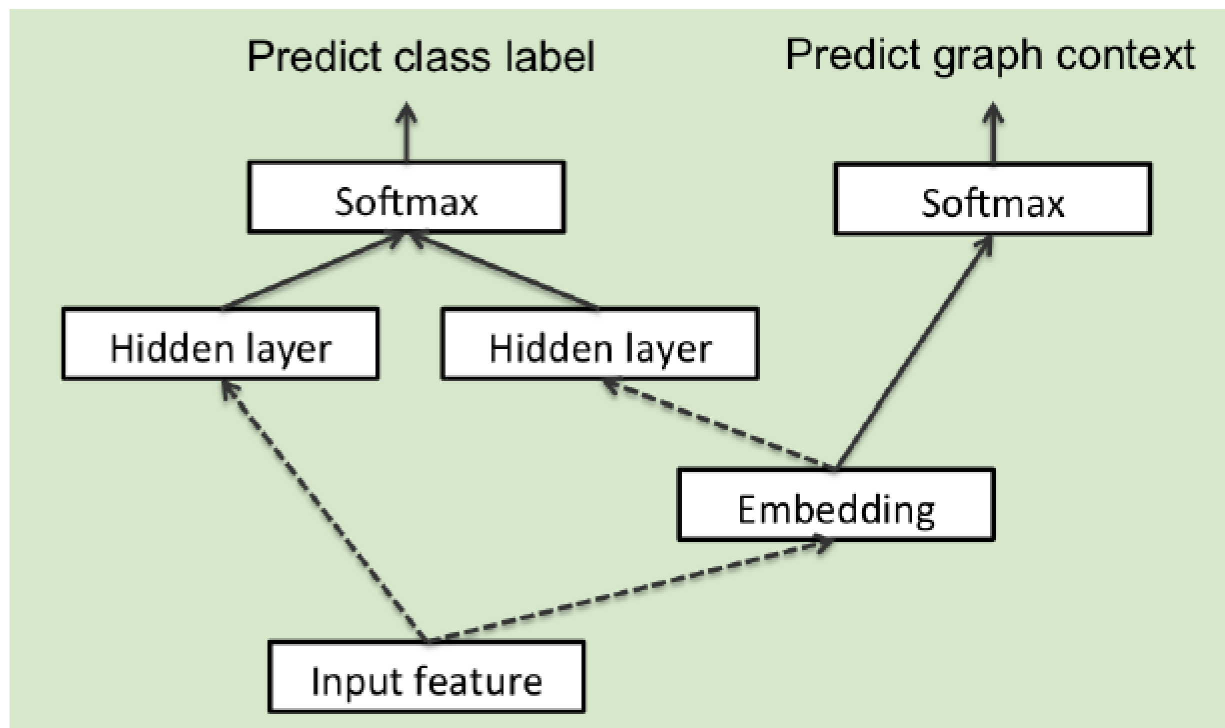
Transductive learning only aims to apply the classifier f on the unlabeled instances observed at training time, and the classifier does not generalize to unobserved instances.

Inductive learning

on the other hand, aims to learn a parameterized classifier f that is generalizable to unobserved instances.



(a) Transductive Formulation



(b) Inductive Formulation

What Is Graph-Based Semi-Supervised Learning?

In addition to labeled and unlabeled instances, a graph, denoted as a $(L+U) \times (L+U)$ matrix A, is also given to graph-based semi-supervised learning methods.

What Is Graph-Based Semi-Supervised Learning?

In addition to labeled and unlabeled instances, a graph, denoted as a $(L + U) \times (L + U)$ matrix A, is also given to graph-based semi-supervised learning methods.

Each entry a_{ij} indicates the similarity between instance i and j, which can be either labeled or unlabeled.

What Is Graph-Based Semi-Supervised Learning?

In addition to labeled and unlabeled instances, a graph, denoted as a $(L + U) \times (L + U)$ matrix A, is also given to graph-based semi-supervised learning methods.

Each entry a_{ij} indicates the similarity between instance i and j, which can be either labeled or unlabeled.

In this paper,
we mainly focus on the setting that a graph is explicitly given and represents additional information not present in the feature vectors

(e.g., the graph edges correspond to hy- perlinks between documents).

The Loss Function

Graph-based semi-supervised learning defines the loss function as a weighted sum of the supervised loss over labeled instances and a graph Laplacian regularization term

The Loss Function

Graph-based semi-supervised learning defines the loss function as a weighted sum of the supervised loss over labeled instances and a graph Laplacian regularization term

The graph Laplacian regularization is based on the assumption that nearby nodes in a graph are likely to have the same labels.

The Loss Function

Graph-based semi-supervised learning defines the loss function as a weighted sum of the supervised loss over labeled instances and a graph Laplacian regularization term

The graph Laplacian regularization is based on the assumption that nearby nodes in a graph are likely to have the same labels.

Generally, the loss function of graph-based semi-supervised learning in the binary case can be written as

$$
\begin{aligned}
& \sum_{i=1}^{L} l(y_i, f(x_i)) + \lambda \sum_{i,j} a_{ij} \| f(x_i) - f(x_j) \|^2 \\
= \; & \sum_{i=1}^{L} l(y_i, f(x_i)) + \lambda \mathbf{f}^T \Delta \mathbf{f} \qquad (1)
\end{aligned}
$$

The Loss Function

Graph-based semi-supervised learning defines the loss function as a weighted sum of the supervised loss over labeled instances and a graph Laplacian regularization term

The graph Laplacian regularization is based on the assumption that nearby nodes in a graph are likely to have the same labels.

$$\begin{aligned} & \sum_{i=1}^{L} l(y_i, f(x_i)) + \lambda \sum_{i,j} a_{ij} \|f(x_i) - f(x_j)\|^2 \\ = & \sum_{i=1}^{L} l(y_i, f(x_i)) + \lambda \mathbf{f}^T \Delta \mathbf{f} \qquad (1) \end{aligned}$$

The graph Laplacian matrix Δ

$\Delta = A - D$, where D is a diagonal matrix with each entry defined as $d_{ii} = \sum_j a_{ij}$.

The Loss Function

Graph-based semi-supervised learning defines the loss function as a weighted sum of the supervised loss over labeled instances and a graph Laplacian regularization term

The graph Laplacian regularization is based on the assumption that nearby nodes in a graph are likely to have the same labels.

$$
\begin{aligned}
& \sum_{i=1}^{L} l(y_i, f(x_i)) + \lambda \underline{\sum_{i,j} a_{ij} \|f(x_i) - f(x_j)\|^2} \\
= \quad & \sum_{i=1}^{L} l(y_i, f(x_i)) + \lambda \underline{\mathbf{f}^T \Delta \mathbf{f}} \qquad \text{How?} \qquad (1)
\end{aligned}
$$

The graph Laplacian matrix Δ

$\Delta = A - D$, where D is a diagonal matrix with each entry defined as $d_{ii} = \sum_j a_{ij}$.

How?

$$\sum_{i,j} a_{i,j} \|f(x_i)-f(x_j)\|^2 =$$

$$\sum_{i,j} a_{i,j} \left(\; f(x_i)^2 + f(x_j)^2 - 2 f(x_i) f(x_j) \; \right) =$$

$$\sum_{i,j} a_{i,j} f(x_i)^2 + \sum_{i,j} a_{i,j} f(x_j)^2 + \sum_{i,j} -2 f(x_i) f(x_j) a_{i,j} =$$

$$\sum_{i,j} a_{i,j} f(x_i)^2 + \sum_{j,i} a_{j,i} f(x_j)^2 + \sum_{i,j} -2 f(x_i) f(x_j) a_{i,j} =$$

$$2 [\sum_{i,j} a_{i,j} f(x_i)^2 + \sum_{i,j} - f(x_i) f(x_j) a_{i,j}] =$$

$$(ignore\, 2)$$

$$\sum_{i} f(x_i)^2 \sum_{j} a_{i,j} - \sum_{i} f(x_i) \sum_{j} f(x_j) a_{i,j} =$$

$$\sum_{i} f(x_i)^2 d_{ii} - \sum_{i} f(x_i) \sum_{j} f(x_j) a_{i,j} =$$

$$f^T D f - f^T A f =$$

$$f^T (D-A) f$$

Semi-Supervised Learning with Graph Embeddings

We formulate our framework based on feed-forward neural networks.

Semi-Supervised Learning with Graph Embeddings

We formulate our framework based on feed-forward neural networks.

Given the input feature vector $\mathbf{x}$, the k-th hidden layer of the network is denoted as $\mathbf{h}^k$, which is a nonlinear function of the previous hidden layer $\mathbf{h}^{k-1}$ defined as: $\mathbf{h}^k(\mathbf{x}) = \mathrm{ReLU}(\mathbf{W}^k \mathbf{h}^{k-1}(\mathbf{x}) + b^k)$, where $\mathbf{W}^k$ and b^k are parameters of the k-th layer, and $\mathbf{h}^0(\mathbf{x}) = \mathbf{x}$. We adopt rectified linear unit $\mathrm{ReLU}(x) = \max(0, x)$ as the nonlinear function in this work.

Semi-Supervised Learning with Graph Embeddings

We formulate our framework based on feed-forward neural networks.

Given the input feature vector $\mathbf{x}$, the k-th hidden layer of the network is denoted as $\mathbf{h}^k$, which is a nonlinear function of the previous hidden layer $\mathbf{h}^{k-1}$ defined as: $\mathbf{h}^k(\mathbf{x}) = \mathrm{ReLU}(\mathbf{W}^k \mathbf{h}^{k-1}(\mathbf{x}) + b^k)$, where $\mathbf{W}^k$ and b^k are parameters of the k-th layer, and $\mathbf{h}^0(\mathbf{x}) = \mathbf{x}$. We adopt rectified linear unit $\mathrm{ReLU}(x) = \max(0, x)$ as the nonlinear function in this work.

The loss function of our framework can be expressed as

$$\mathcal{L}_s + \lambda \mathcal{L}_u,$$

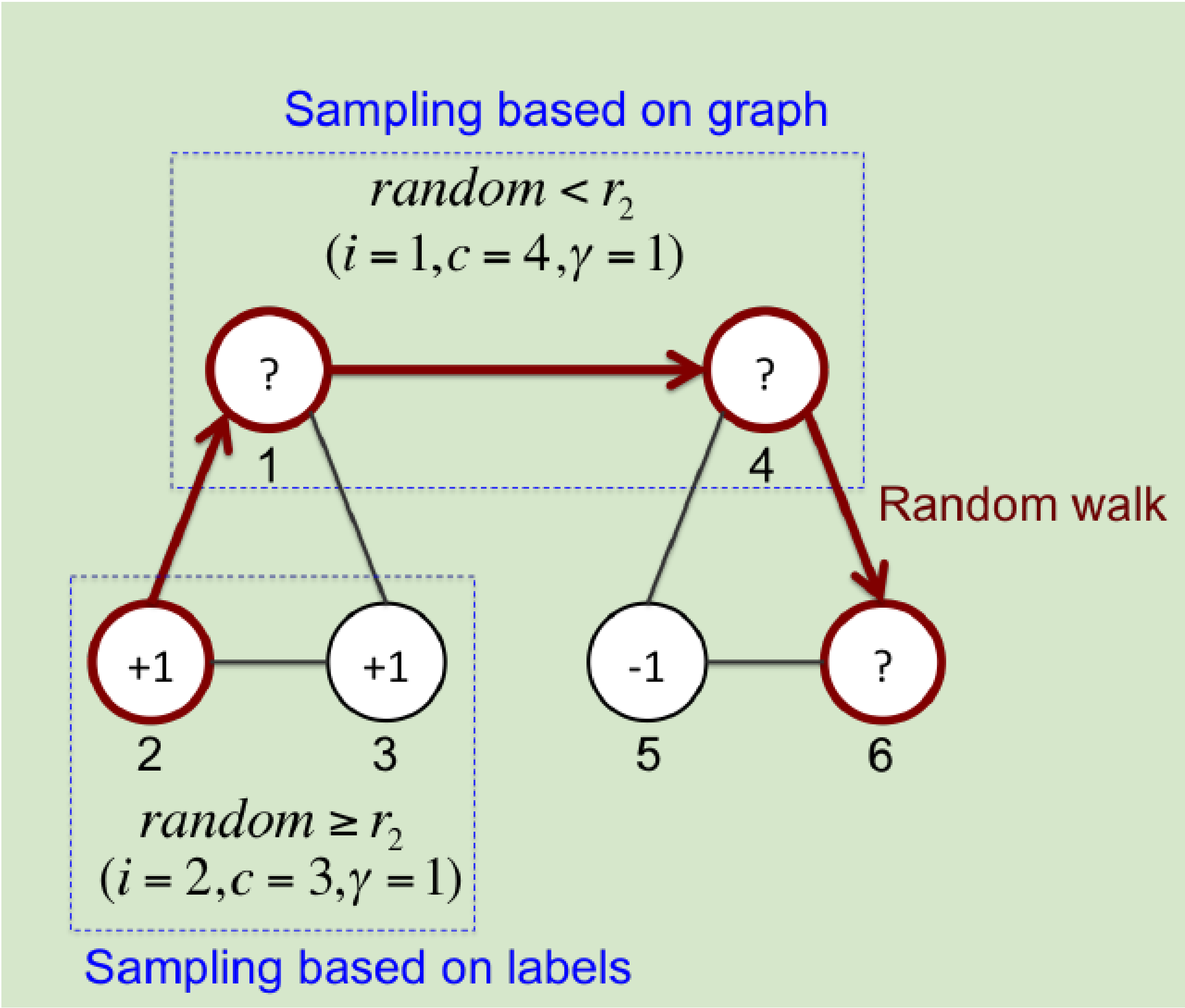
where $\mathcal{L}_s$ is a supervised loss of predicting the labels, and $\mathcal{L}_u$ is an unsupervised loss of predicting the graph context.

Unsupervised Loss

the unsupervised loss with negative sampling can be written as

$$\mathcal{L}_u = -\mathbb{E}_{(i,c,\gamma)} \log \sigma(\gamma \mathbf{w}_c^T \mathbf{e}_i) \quad (3)$$

Unsupervised Loss



Unsupervised Loss

Algorithm 1 Sampling Context Distribution $p(i, c, \gamma)$

Input: graph A, labels $y_{1:L}$, parameters r_1, r_2, q, d
Initialize triplet (i, c, γ)
if $random < r_1$ **then** $\gamma \leftarrow +1$ **else** $\gamma \leftarrow -1$
if $random < r_2$ **then**
  Uniformly sample a random walk S of length q
  Uniformly sample (S_j, S_k) with $|j - k| < d$
  $i \leftarrow S_j, c \leftarrow S_k$
  if $\gamma = -1$ **then** uniformly sample c from $1 : L + U$
else
  if $\gamma = +1$ **then**
    Uniformly sample (i, c) with $y_i = y_c$
  else
    Uniformly sample (i, c) with $y_i \neq y_c$
  end if
end if
return (i, c, γ)

Label Loss

$$p(y|\mathbf{x},\mathbf{e}) = \frac{\exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{e})^T]\mathbf{w}_y}{\sum_{y'} \exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{e})^T]\mathbf{w}_{y'}}, \qquad (4)$$

$[\cdot, \cdot]$ denotes concatenation of two row vectors and $\mathbf{w}$ represents the model parameter.

Label Loss

$$p(y|\mathbf{x}, \mathbf{e}) = \frac{\exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{e})^T]\mathbf{w}_y}{\sum_{y'} \exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{e})^T]\mathbf{w}_{y'}}, \qquad (4)$$

Combined with Eq. (3), the loss function of transductive learning is defined as:

$$-\frac{1}{L}\sum_{i=1}^{L} \log p(y_i|\mathbf{x}_i, \mathbf{e}_i) - \lambda \mathbb{E}_{(i,c,\gamma)} \log \sigma(\gamma \mathbf{w}_c^T \mathbf{e}_i),$$

Inductive Formulation

Replacing $\mathbf{e}_i$ in Eq. (3) with $\mathbf{h}^{l_1}(\mathbf{x}_i)$, the loss function of inductive learning is

$$-\frac{1}{L}\sum_{i=1}^{L}\log p(y_i|\underline{\mathbf{x}_i}) - \lambda\mathbb{E}_{(i,c,\gamma)}\log\sigma(\gamma\mathbf{w}_c^T\underline{\mathbf{h}^{l_1}(\mathbf{x}_i)})$$

Model Training

Algorithm 2 Model Training (Transductive)

Input: A, $\mathbf{x}_{1:L+U}$, $y_{1:L}$, λ, batch iterations T_1, T_2 and sizes N_1, N_2

repeat

  for $t \leftarrow 1$ **to** T_1 **do**

    Sample a batch of labeled instances i of size N_1

    $\mathcal{L}_s = -\frac{1}{N_1}\sum_i p(y_i|\mathbf{x}_i, \mathbf{e}_i)$

    Take a gradient step for $\mathcal{L}_s$

  end for

  for $t \leftarrow 1$ **to** T_2 **do**

    Sample a batch of context from $p(i, c, \gamma)$ of size N_2

    $\mathcal{L}_u = -\frac{1}{N_2}\sum_{(i,c,\gamma)} \log \sigma(\gamma \mathbf{w}_c^T \mathbf{e}_i)$

    Take a gradient step for $\mathcal{L}_u$

  end for

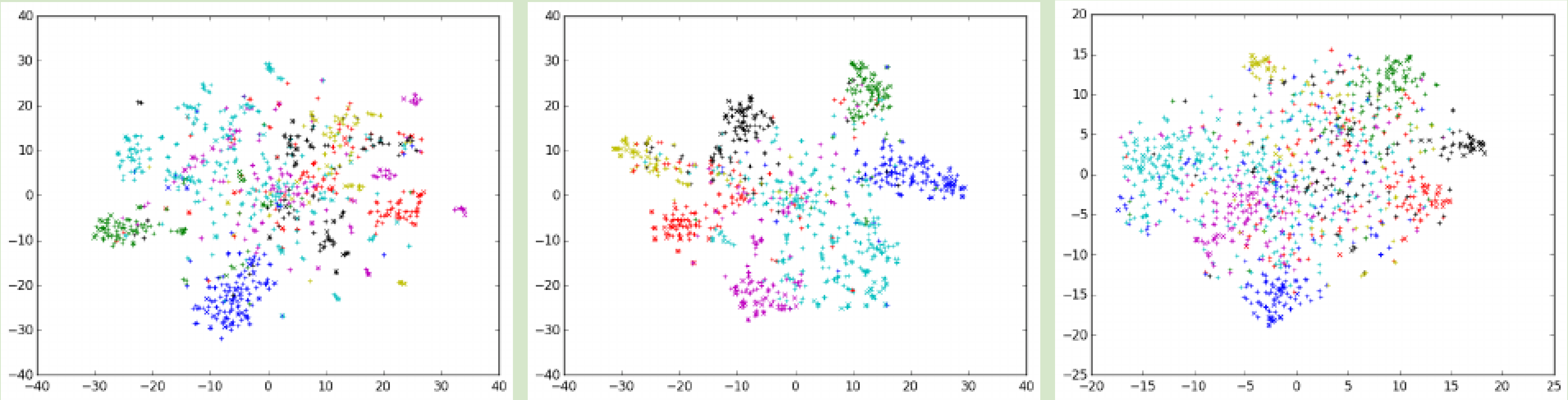
until stopping

Supervised Training on label

SGD with mini-batch mode

Unsupervised Training on context

Experiments



(a) GraphEmb (b) Planetoid-T (c) SemiEmb

Figure 3. t-SNE Visualization of embedding spaces on the Cora dataset. Each color denotes a class.

One More Thing

Thanks for Listening

Preferences

- [1] 《Revisiting Semi-supervised Learning with Graph Embeddings. Zhilin Yang, William W. Cohen, Ruslan Salakhutdinov. ICML 2016.》
- [2]