Paper: Mixup for Node and Graph Classification [2]

Jiaxin Liu

Group Reading

July 12, 2021

Overview

- Mix-up example

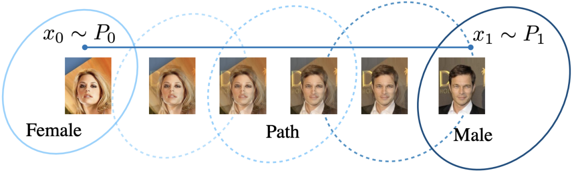


Figure: Example for interpolation.

From ERM to VRM [1]

- Expected Risk:
  - Function $f : X \mapsto Y$ where $(x, y) \sim P(X, Y)$.
  - Loss function $l : X \times Y \to \mathbb{R}$
  - $R(f) = \int l(f(x), y) dP(x, y)$

From ERM to VRM [1]

- Expected Risk:
  - Function $f: X \mapsto Y$ where $(x, y) \sim P(X, Y)$.
  - Loss function $l: X \times Y \to \mathbb{R}$
  - $R(f) = \int l(f(x), y) dP(x, y)$
- Empirical Risk:
  - Training data $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $(x_i, y_i) \sim P$ for all $i = 1, \ldots, n$.
  - $R_\delta(f) = \int l(f(x), y) dP_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i)$
- Pros and Cons? Other methods to approximate P? Vicinal Risk Minimization (VRM) [1].

From VRM to Mixup

- Vicinity distribution $P_v(\tilde{x}, \tilde{y}) = \frac{1}{n}\sum_{i=1}^{n} v(\tilde{x}, \tilde{y}|x_i, y_i)$
  - $v(\tilde{x}, \tilde{y}|x_i, y_i) = \mathcal{N}(\tilde{x} - x_i, \sigma^2)\delta(\tilde{y} = y_i)$
  - Dataset $D_v := \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{m}$
  - Empirical vicinal risk: $R_v(f) = \frac{1}{m}\sum_{i=1}^{m} l(f(\tilde{x}_i), \tilde{y}_i)$

From VRM to Mixup

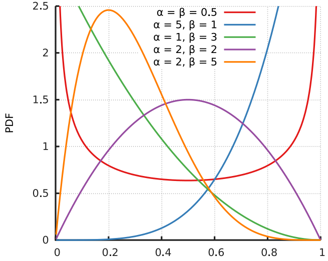
- Vicinity distribution $P_v(\tilde{x},\tilde{y}) = \frac{1}{n}\sum_{i=1}^{n} v(\tilde{x},\tilde{y}|x_i,y_i)$
  - $v(\tilde{x},\tilde{y}|x_i,y_i) = \mathcal{N}(\tilde{x}-x_i,\sigma^2)\delta(\tilde{y}=y_i)$
  - Dataset $D_v := \{(\tilde{x}_i,\tilde{y}_i)\}_{i=1}^{m}$
  - Empirical vicinal risk: $R_v(f) = \frac{1}{m}\sum_{i=1}^{m} l(f(\tilde{x}_i),\tilde{y}_i)$
- Mixup
  - $\mu(\tilde{x},\tilde{y}|x_i,y_i) = \frac{1}{n}\sum_j^n \mathbb{E}_\lambda[\delta(\tilde{x}=\lambda\cdot x_i+(1-\lambda)\cdot x_j, \tilde{y}=\lambda\cdot y_i+(1-\lambda)\cdot y_j)]$ where $\lambda \sim \mathrm{Beta}(\alpha,\alpha)$, for $\alpha\in(0,\infty)$.
  - $\tilde{x}=\lambda x_i+(1-\lambda)x_j$, $\tilde{y}=\lambda y_i+(1-\lambda)y_j$ where $\lambda\in[0,1]$
  - 

Figure: Beta distributions.

Mixup Illustration

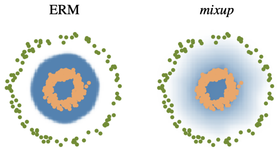


Figure: Toy example. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y=1|x)$. [3]

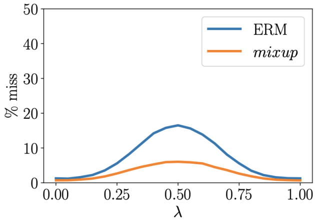


Figure: Prediction error in-between training data. Evaluated at $x = \lambda x_i + (1-\lambda)x_j$, a prediction is counted as a "miss" if it does not belong to $\{y_i, y_j\}$. The model trained with mixup has fewer misses. [3]

Mixup for Graph

- Graph $G = (V, E)$
- x_i, neighborhood of node i is $N(i) = \{j \in V | (i,j) \in E\}$
- GNN:

$$h_i^{(l)} = \text{AGGREGATE}(h_i^{(l-1)}, \{h_j^{(l-1)} | j \in N(i)\}, W^{(l)}).$$

$$h_i^{(0)} = x_i.$$

- Graph classification: $h_G = \text{READOUT}(\{h_i^{(L)} | i \in V\})$.

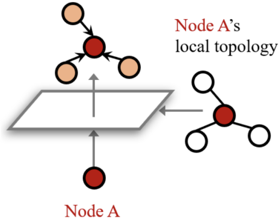


Figure: A GNN layer.

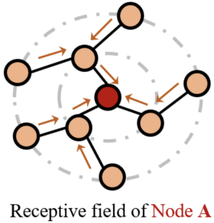


Figure: Receptive field of node A.

Mixup for Node Classification

- Mixup: $\tilde{x}_{i,j} = \lambda x_i + (1-\lambda)x_j$,
- Two-branch Mixup for nodes (mix the receptive field subgraphs):

$$\tilde{h}^{(l)}_{ij,i} = \text{AGGREGATE}(\tilde{h}^{(l-1)}_{ij}, \{h^{(l-1)}_k | k \in N(i)\}, W^{(l)})$$

$$\tilde{h}^{(l)}_{ij,j} = \text{AGGREGATE}(\tilde{h}^{(l-1)}_{ij}, \{h^{(l-1)}_k | k \in N(j)\}, W^{(l)})$$

Node mixup

$$\tilde{h}^{(l)}_{ij} = \lambda \tilde{h}^{(l)}_{ij,i} + (1-\lambda)\tilde{h}^{(l)}_{ij,j}.$$

where $\tilde{h}^{(0)}_{ij} = \tilde{x}_{i,j}$.

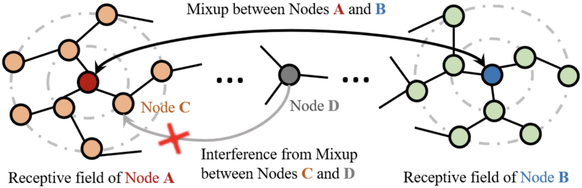


Figure: Two-branch mixup for nodes A and B.

Mixup for Node Classification

How to get $h_k^{(l)}$ and $\tilde{h}_{ij}^{(l)}$? Two-stage Mixup.

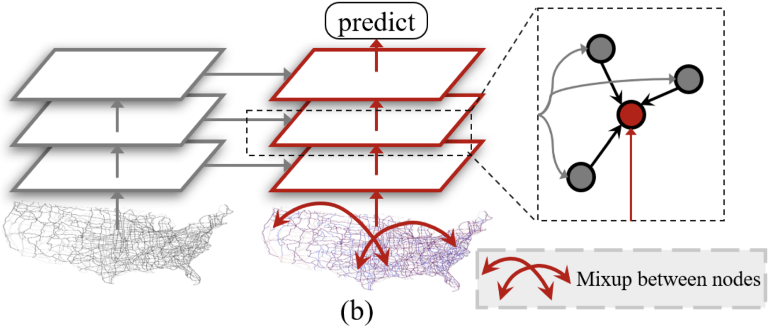


Figure: Two-stage mixup for getting $h_k^{(l)}$ and $\tilde{h}_{ij}^{(l)}$.

Mixup for Graph Classification

- Only mixup two graphs in the embedding space.

$$\tilde{h}_{G_1,G_2} = \lambda h_{G_1} + (1-\lambda) h_{G_2}$$

$$\tilde{y}_{G_1,G_2} = \lambda y_{G_1} + (1-\lambda) y_{G_2}$$

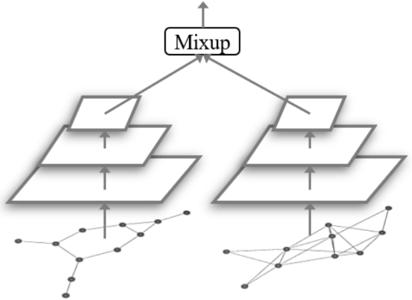


Figure: Mixup for graph classification.

Algorithm

Algorithm 1 Two-Stage Mixup for Node Classification

Input: Graph $G = (\mathcal{V}, \mathcal{E})$ of a mini-batch, with node attributes $\{\mathbf{x}_i | i \in \mathcal{V}\}$, a GNN model with the aggregation function AGGREGATE$(\cdot)$, hyper-parameter α for the distribution of λ, the ground truth labels $\{\mathbf{y}_i | i \in \mathcal{V}\}$.

Output: The trained parameters of GNN: $\left\{\mathbf{W}^{(l)}\right\}_l$.

Stage 1

for $i \leftarrow 1$ to $\#\mathcal{V}$ **do**
    $\mathbf{h}_i^{(0)} \leftarrow \mathbf{x}_i$
end for
for $l \leftarrow 1$ to $L - 1$ **do**
    for $i \leftarrow 1$ to $\#\mathcal{V}$ **do**
        $\mathbf{h}_i^{(l)} \leftarrow \text{AGGREGATE}\left(\mathbf{h}_i^{(l-1)}, \left\{\mathbf{h}_j^{(l-1)} | j \in \mathcal{N}(i)\right\}, \mathbf{W}^{(l)}\right)$
    end for
end for

Stage 2

for $i \leftarrow 1$ to $\#\mathcal{V}$ **do**
    Sample j from $\mathcal{V}$
    $\lambda \leftarrow \text{Beta}(\alpha, \alpha)$
    $\tilde{\mathbf{x}}_{ij} \leftarrow \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j$
    $\tilde{\mathbf{y}}_{ij} \leftarrow \lambda \mathbf{y}_i + (1 - \lambda)\mathbf{y}_j$
    $\tilde{\mathbf{h}}_{ij}^{(0)} \leftarrow \tilde{\mathbf{x}}_{ij}$
    for $l \leftarrow 1$ to L **do**
        $\tilde{\mathbf{h}}_{ij,i}^{(l)} \leftarrow \text{AGGREGATE}\left(\tilde{\mathbf{h}}_{ij}^{(l-1)}, \left\{\mathbf{h}_k^{(l-1)} | k \in \mathcal{N}(i)\right\}, \mathbf{W}^{(l)}\right)$ → Branch 1
        $\tilde{\mathbf{h}}_{ij,j}^{(l)} \leftarrow \text{AGGREGATE}\left(\tilde{\mathbf{h}}_{ij}^{(l-1)}, \left\{\mathbf{h}_k^{(l-1)} | k \in \mathcal{N}(j)\right\}, \mathbf{W}^{(l)}\right)$ → Branch 2
        $\tilde{\mathbf{h}}_{ij}^{(l)} \leftarrow \lambda \tilde{\mathbf{h}}_{ij,i}^{(l)} + (1 - \lambda)\tilde{\mathbf{h}}_{ij,j}^{(l)}$
    end for
end for

Calculate classification loss $\mathcal{L}$ on $\left\{\tilde{\mathbf{h}}_{ij}^{(L)}, \tilde{\mathbf{y}}_{ij} | i \in \mathcal{V}\right\}$.
Back-propagation on $\left\{\mathbf{W}^{(l)}\right\}_l$ for minimizing $\mathcal{L}$.

Experiments

- Node classification

Method	Citeseer	Cora	Pubmed
GCN [27]	77.1±1.4	88.3±0.8	86.4±1.1
GAT [50]	76.3±0.8	87.6±0.5	85.7±0.7
JKNet [61]	78.1±0.9	89.1±1.2	86.9±1.3
LGCN [18]	77.5±1.1	89.0±1.2	86.5±0.6
GMNN [39]	77.4±1.5	88.7±0.8	86.7±1.0
ResGCN [31]	77.9±0.8	88.1±0.6	87.1±1.2
DropEdge [40] + GCN	78.1±1.1	89.2±0.7	87.3±0.6
DropEdge [40] + JKNet	79.3±0.7	89.9±0.8	87.6±0.9
Mixup + GCN	78.7±0.9	90.0±0.7	87.9±0.8
Mixup + JKNet	**80.1±0.8**	**90.4±0.9**	**88.3±0.6**

Figure: Test Accuracy of transductive node classification.

Experiments

- Varied training ratios r.

Method	Citeseer $r = 30\%$	Citeseer $r = 40\%$	Citeseer $r = 50\%$	Cora $r = 30\%$	Cora $r = 40\%$	Cora $r = 50\%$	Pubmed $r = 30\%$	Pubmed $r = 40\%$	Pubmed $r = 50\%$
GCN [27]	74.7 ± 2.5	75.2 ± 1.8	76.3 ± 1.6	86.3 ± 1.9	86.8 ± 1.4	87.5 ± 1.0	85.1 ± 2.3	85.4 ± 1.4	85.8 ± 1.2
Mixup + GCN	76.9 ± 2.1	77.1 ± 1.5	78.1 ± 1.3	88.5 ± 1.4	88.9 ± 1.0	89.4 ± 0.9	87.0 ± 1.6	87.2 ± 1.1	87.5 ± 1.0
JKNet [61]	75.6 ± 1.9	76.0 ± 1.4	77.1 ± 1.1	86.7 ± 2.1	87.4 ± 1.5	88.2 ± 1.3	85.3 ± 2.2	85.9 ± 1.6	86.4 ± 1.4
Mixup + JKNet	**78.0 ± 1.7**	**78.3 ± 1.2**	**79.2 ± 1.0**	**88.6 ± 2.0**	**89.1 ± 1.5**	**89.7 ± 1.2**	**87.2 ± 1.9**	**87.5 ± 1.3**	**87.9 ± 0.9**

Figure: Test Accuracy of node classification over different training set ratios.

Experiments

- t-SNE plot for the final-layer representations.
- Loss on the test data during training.

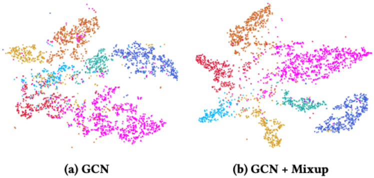


Figure: The learned representations of the nodes in the Cora dataset.

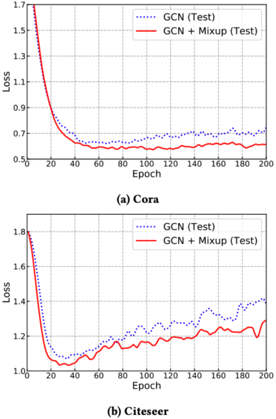


Figure: The training curves of GCN with and without Mixup.

Experiments

- Two-stage framework.

Method	two stages	Pubmed	Δ	Yelp	Δ
GCN [27]	-	86.4±1.1	0	65.3±0.3	0
Mixup + GCN	w/o	85.8±1.3	-0.6	64.2±0.6	-1.1
	w/	87.9±0.8	+1.5	66.3±0.4	+1.0

Figure: The node classification results with and without two-stage framework.

- Selection for α in Beta(α, α).

Method	α	Citeseer	Cora	Flickr
GCN [27]	-	77.1±1.4	88.3±0.8	51.1±0.2
Mixup + GCN	0.2	78.1±0.9	89.2±0.8	52.0±0.3
	0.5	78.4±0.8	89.5±0.7	52.1±0.3
	1	**78.7±0.9**	**90.0±0.7**	52.4±0.4
	2	78.6±1.0	89.8±0.8	**52.8±0.5**
	5	78.4±1.2	89.4±1.1	52.7±0.4

Figure: Node classification results with different

- Olivier Chapelle et al. “Vicinal risk minimization”. In: *Advances in neural information processing systems* (2001), pp. 416–422.
- Yiwei Wang et al. “Mixup for Node and Graph Classification”. In: *Proceedings of the Web Conference 2021*. 2021, pp. 3663–3674.
- Hongyi Zhang et al. “mixup: Beyond empirical risk minimization”. In: *arXiv preprint arXiv:1710.09412* (2017).