Label Propagation for Deep Semi-supervised Learning

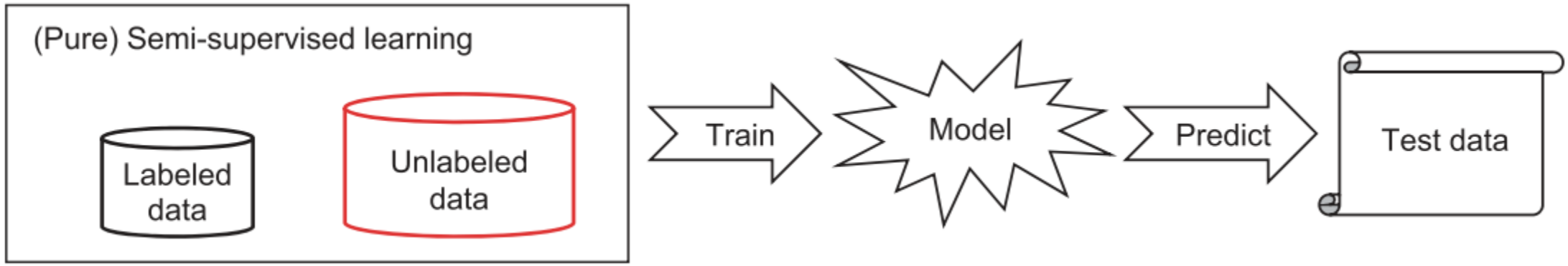
Ahmet Iscen, Giorgos Tolias, Yannis Avrithis & Ondrej Chum

Czech Technical University in Prague & Univ Rennes, Inria, CNRS, IRISA

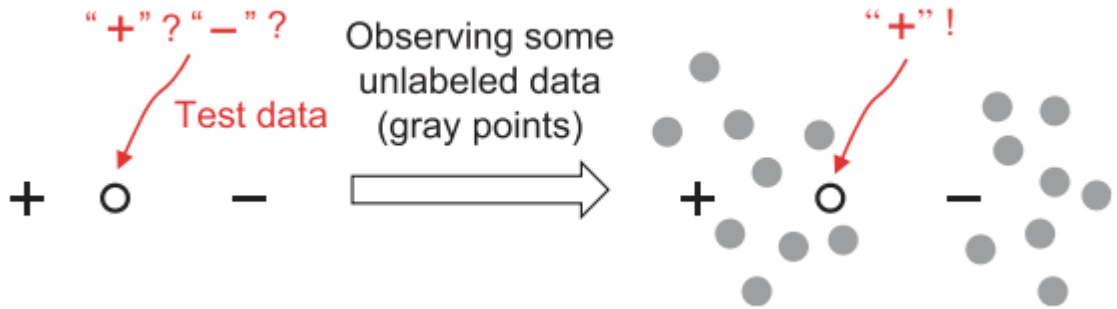
Presented by Xun Xu

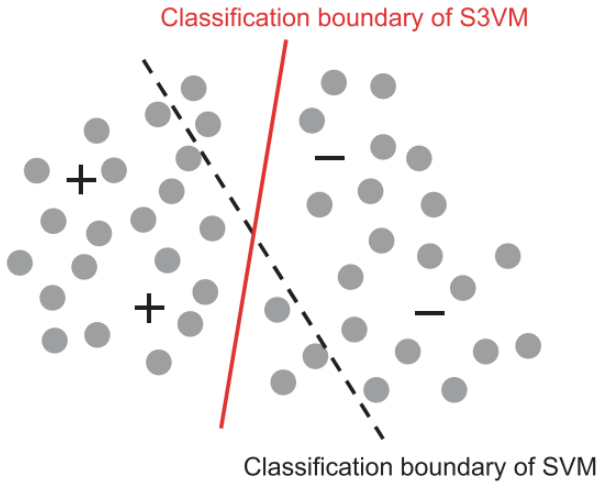
Motivations – What is Semi-Supervised Learning (SSL) [1]

- Objective of Semi-Supervised Learning



- Why it works?

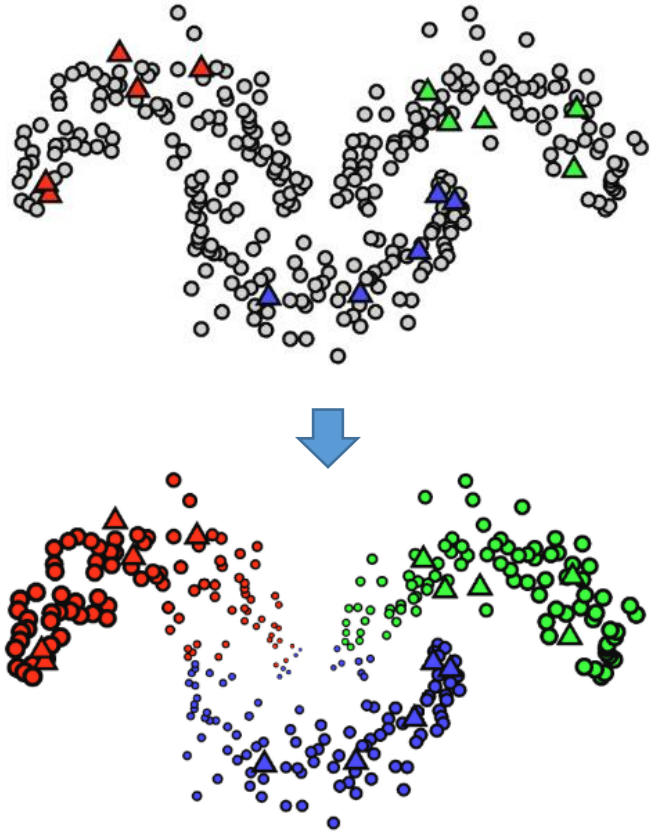




[1] Zhou, Zhi-Hua. "A brief introduction to weakly supervised learning." *National Science Review* 5.1 (2017): 44-53.

Motivation - Examples

- Examples of SSL on toy example & computer vision data



Point-level

Only a few points are labelled

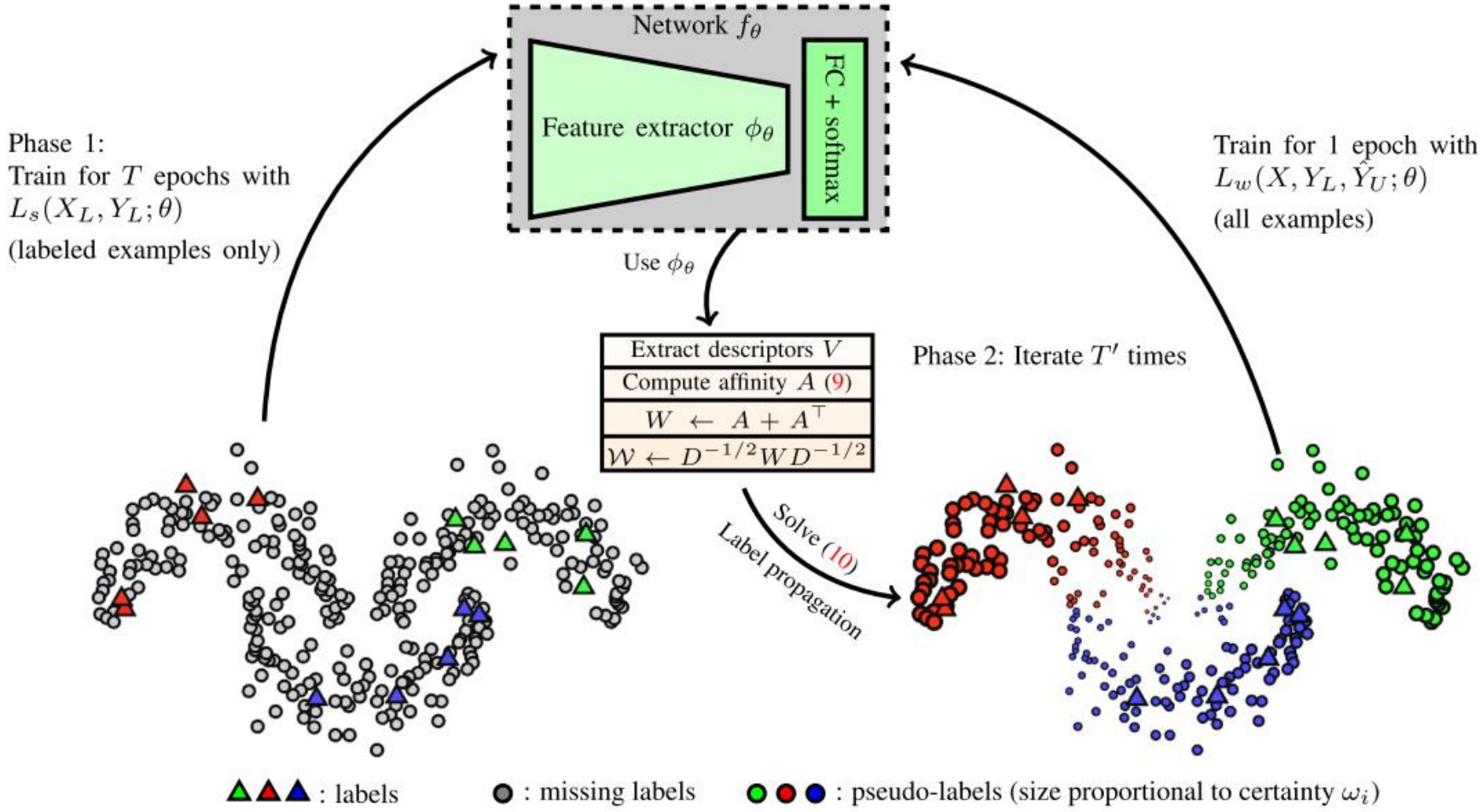
Infer the full labels

Problem Formulation

- Notations
- Input examples: $\mathbf{X} = \{\mathbf{x}_1, \cdots \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots \mathbf{x}_n\}$
- Input labels: $\mathbf{Y}_l = \{y_1, \cdots y_l\}, \quad y_i \in \{1, \cdots, C\}$
- Labelled examples: $\mathbf{X}_l = \{\mathbf{x}_i \mid i \in \{1, \cdots l\}\}$
- Unlabelled examples: $\mathbf{X}_u = \{\mathbf{x}_i \mid i \in \{l+1, \cdots n\}\}$
- Labelled/Unlabelled Set: $\mathbf{L} = \{1, \cdots, l\} \quad \mathbf{U} = \{l+1, \cdots, n\}$
- Adjacency/Weight Matrix: $\mathbf{W} \in \mathbf{R}^{N \times N}$
- Degree Matrix: $\mathbf{D} = diag(\mathbf{d}) \quad d_i = \sum_j w_{ij}$

Overview of Algorithm

- Algorithm overview for self-supervised label propagation



K Nearest Neighbor Graph

- Encoder network

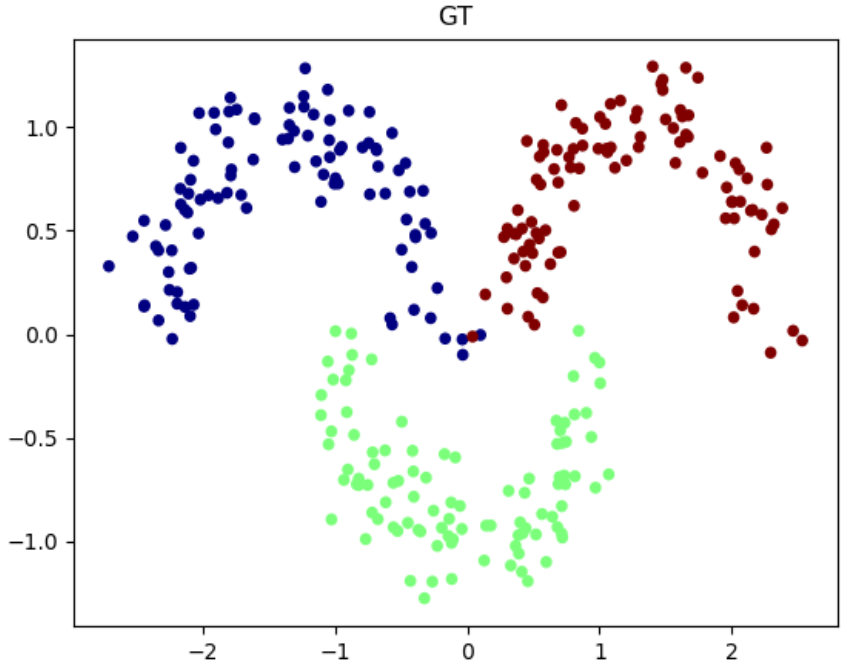
$$\mathbf{v}_i = \phi(\mathbf{x}_i; \boldsymbol{\theta})$$

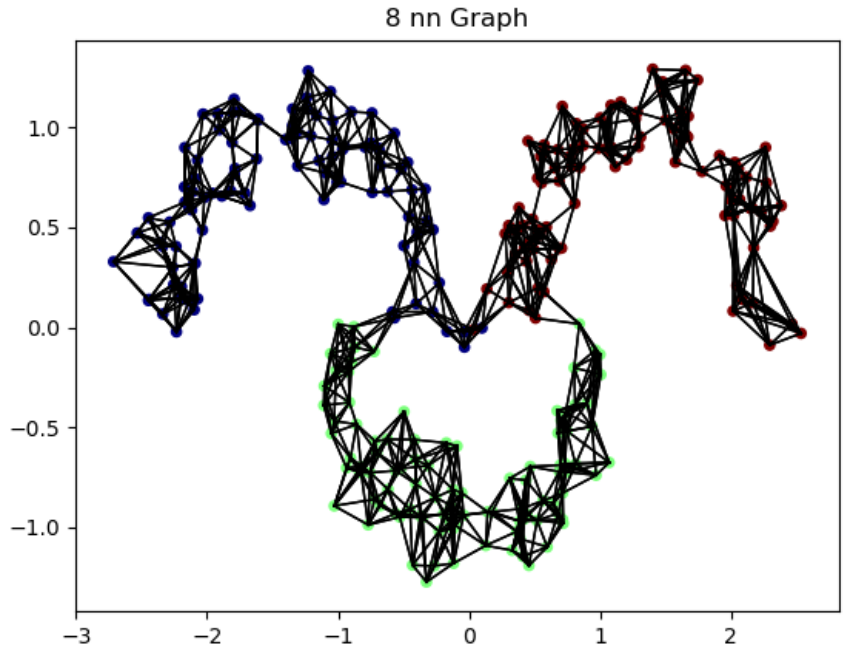
- Knn Graph

$$a_{ij} = \begin{cases} \exp\left(-\mathbf{v}_i^T \mathbf{v}_j / \lambda\right) & \mathbf{v}_j \in NN_k(\mathbf{v}_i) \\ 0 & otherwise \end{cases}$$

- Symmetric Knn Graph

$$\mathbf{W} = \mathbf{A} + \mathbf{A}^T$$





Label Propagation

- Normalize Adajency Matrix $\tilde{\mathbf{W}} = \mathbf{D}^{-0.5}\mathbf{W}\mathbf{D}^{-0.5}$
- One-hot Encoded Label Matrix $y_{ij} = \begin{cases} 1 & i \in \{1, \cdots l\} \wedge y_i = j \\ 0 & otherwise \end{cases}, \quad Y \in \{0,1\}^{l+u \times C}$
- Closed-form Label Predictions $\mathbf{Z} = \left(\mathbf{I} - \alpha\tilde{\mathbf{W}}\right)^{-1}\mathbf{Y}$ Eq(6) in the paper
- Class Prediction $\tilde{y}_i = \arg\max_j z_{ij}$

Label Propagation 15 Years Ago: An Algorithmic Perspective [1,2]

- Label Propagation Algorithm
  - Step 1 Compute normalized weight matrix $\tilde{\mathbf{W}} = \mathbf{D}^{-0.5}\mathbf{W}\mathbf{D}^{-0.5}$
  - Step 2 Propagate labels according to rule $\mathbf{Z}^{(t+1)} = \alpha\tilde{\mathbf{W}}\mathbf{Z}^{t} + (1-\alpha)\mathbf{Y}$
  - Step 3 Repeat previous step until convergence
  - Step 4 Assign labels $y_i = \arg\max_j \mathbf{Z}_{ij}^{\infty}$

Adjacency Matrix $\mathbf{W}$

Balance Param. $\alpha \in [0,1]$

Initial Labels $\mathbf{Y}$

Propagated Labels $\mathbf{Z}$

- Closed-Form Solution for Step 2 & Step 3

$$\mathbf{Z}^{\infty} = \alpha\tilde{\mathbf{W}}\left(\alpha\tilde{\mathbf{W}}\left(\alpha\tilde{\mathbf{W}}(\cdots)+(1-\alpha)\mathbf{Y}\right)+(1-\alpha)\mathbf{Y}\right)+(1-\alpha)\mathbf{Y}$$

$$= \alpha^{\infty}\tilde{\mathbf{W}}^{\infty}\mathbf{Z}^{0} + \sum_{m=0\cdots\infty}\alpha^{m}\tilde{\mathbf{W}}^{m}(1-\alpha)\mathbf{Y} = \sum_{m=0\cdots\infty}\alpha^{m}\tilde{\mathbf{W}}^{m}(1-\alpha)\mathbf{Y}$$

$$= (1-\alpha)\left(\mathbf{I}-\alpha\tilde{\mathbf{W}}\right)^{-1}\mathbf{Y} \propto \left(\mathbf{I}-\alpha\tilde{\mathbf{W}}\right)^{-1}\mathbf{Y}$$

[1] Zhu, Xiaojin, Zoubin Ghahramani, and John D. Lafferty. "Semi-supervised learning using gaussian fields and harmonic functions." *ICML03*

[2] Zhou, Dengyong, et al. "Learning with local and global consistency." NIPS04

Proof of Normalized Matrix's Eigenvalue Bounds

- Normalized Weight Matrix

$$\tilde{\mathbf{W}} = \mathbf{D}^{-0.5}\mathbf{W}\mathbf{D}^{-0.5} \text{ so } \left|eig(\tilde{\mathbf{W}})\right| \le 1$$

Proof: [1]

Laplacian Matrix $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{W}} \succ 0$

$$\forall \mathbf{x} \in \mathbf{R}^N \quad \mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{i=1\cdots N} x_i^2 - \sum_{i,j} \tilde{w}_{ij} x_i x_j = \frac{1}{2}\sum_{i,j} \tilde{w}_{ij}\left(x_i - x_j\right)^2 \ge 0$$

Thus the Rayleigh Quotient is upper bounded by 1

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \mathbf{x}^T\left(\mathbf{I} - \tilde{\mathbf{W}}\right)\mathbf{x} = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\tilde{\mathbf{W}}\mathbf{x} \ge 0 \Rightarrow 1 \ge \frac{\mathbf{x}^T\tilde{\mathbf{W}}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$$

The Rayleigh Quotient is lower bounded by -1

$$\mathbf{x}^T\left(\mathbf{I} + \tilde{\mathbf{W}}\right)\mathbf{x} = \mathbf{x}^T\mathbf{x} + \mathbf{x}^T\tilde{\mathbf{W}}\mathbf{x} \ge 0 \Rightarrow -1 \le \frac{\mathbf{x}^T\tilde{\mathbf{W}}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$$

The Rayleigh Quotient is bounded by the minimal and maximal eigenvalues of $\tilde{\mathbf{W}}$

[1] https://people.orie.cornell.edu/dpw/orie6334/lecture7.pdf

Proof of Rayleigh Quotient Bound

- Consider the following problem (minimizing Rayleigh Quotient)

$$\min_{\mathbf{x}} \frac{\mathbf{x}^T \tilde{\mathbf{W}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad s.t. \quad \mathbf{x}^T \mathbf{x} = 1 \Rightarrow \min_{\mathbf{x}} \mathbf{x}^T \tilde{\mathbf{W}} \mathbf{x} \quad s.t. \quad \mathbf{x}^T \mathbf{x} = 1$$

Lagrangian Multiplier $L(\mathbf{x}, \lambda) = \mathbf{x}^T \tilde{\mathbf{W}} \mathbf{x} - \lambda \left(\mathbf{x}^T \mathbf{x} - 1 \right)$

Derivative of Lagragian $\frac{\partial L}{\partial \mathbf{x}} = \tilde{\mathbf{W}} \mathbf{x} - \lambda \mathbf{x} = 0 \Rightarrow \tilde{\mathbf{W}} \mathbf{x} = \lambda \mathbf{x}$

- Stationary Point Only at Eigenvectors (same applied to maximizing Rayleigh Quotient)

$$\lambda_{\min} \le \mathbf{x}^T \tilde{\mathbf{W}} \mathbf{x} \le \lambda_{\max} \quad s.t. \quad \mathbf{x}^T \mathbf{x} = 1$$

$$\Rightarrow \lambda_{\min} \le \frac{\mathbf{x}^T \tilde{\mathbf{W}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \le \lambda_{\max}$$

Proof of Label Propagation Convergence

- Normalized Weight Matrix

  $\tilde{\mathbf{W}} = \mathbf{D}^{-0.5}\mathbf{W}\mathbf{D}^{-0.5}$ so $\left|eig(\tilde{\mathbf{W}})\right| \le 1$

  and $\tilde{\mathbf{W}} = \mathbf{P}\Lambda\mathbf{P}^{-1}$

  where $\Lambda = diag(eig(\tilde{\mathbf{W}}))$

- Solve the summation

$$\mathbf{Z}^{\infty} = \sum_{m=0\cdots\infty} \alpha^m \tilde{\mathbf{W}}^m (1-\alpha)\mathbf{Y}$$

$$\sum_{m=0\cdots\infty} \alpha^m \tilde{\mathbf{W}}^m = \mathbf{I} + \alpha\mathbf{P}\Lambda\mathbf{P}^{-1} + \alpha^2\mathbf{P}\Lambda\mathbf{P}^{-1}\mathbf{P}\Lambda\mathbf{P}^{-1} + \alpha^3\mathbf{P}\Lambda\mathbf{P}^{-1}\mathbf{P}\Lambda\mathbf{P}^{-1}\mathbf{P}\Lambda\mathbf{P}^{-1} + \ldots$$

$$= \mathbf{I} + \alpha\mathbf{P}\Lambda\mathbf{P}^{-1} + \alpha^2\mathbf{P}\Lambda^2\mathbf{P}^{-1} + \alpha^3\mathbf{P}\Lambda^3\mathbf{P}^{-1} + \cdots$$

$$= \mathbf{P}\sum_{m=0\cdots\infty} \alpha^m \Lambda^m \mathbf{P}^{-1}$$

$$= \mathbf{P}\begin{bmatrix} \sum_m (\alpha\lambda_1)^m & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sum_m (\alpha\lambda_N)^m \end{bmatrix}\mathbf{P}^{-1} = \mathbf{P}\begin{bmatrix} \frac{1}{1-\alpha\lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{1-\alpha\lambda_N} \end{bmatrix}\mathbf{P}^{-1}$$

$$= \left(\mathbf{P}\begin{bmatrix} 1-\alpha\lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1-\alpha\lambda_N \end{bmatrix}\mathbf{P}^{-1}\right)^{-1} = \left(\mathbf{I} - \alpha\mathbf{P}\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix}\mathbf{P}^{-1}\right)^{-1}$$

$$= \left(\mathbf{I} - \alpha\tilde{\mathbf{W}}\right)^{-1}$$

Proof of Label Propagation Convergence

- Alternatively, the convergence of label propagation is easily verified as

- Take the limit on both sides

$$\lim_{t\to\infty} \mathbf{Z}^{(t+1)} = \lim_{t\to\infty} \alpha\tilde{\mathbf{W}}\mathbf{Z}^{t} + (1-\alpha)\mathbf{Y}$$

$$\Rightarrow \mathbf{Z}^{\infty} = \alpha\tilde{\mathbf{W}}\mathbf{Z}^{\infty} + (1-\alpha)\mathbf{Y}$$

$$\Rightarrow \left(\mathbf{I} - \alpha\tilde{\mathbf{W}}\right)\mathbf{Z}^{\infty} = (1-\alpha)\mathbf{Y}$$

$$\Rightarrow \mathbf{Z}^{\infty} = (1-\alpha)\left(\mathbf{I} - \alpha\tilde{\mathbf{W}}\right)^{-1}\mathbf{Y}$$

Label Propagation: Optimization Perspective

- The following objective is minimize for the purpose of label propagation

$$\min_{\{\mathbf{z_i}\}} \frac{1}{2}\left(\sum_i \sum_j w_{ij} \left\| \frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}} \right\|_2^2 + \beta \sum_i \left\| \mathbf{z}_i - \mathbf{y}_i \right\|_2^2 \right)$$

Eq(8) in the paper

Smoothness Constraint

Fitting Constraint

- Smoothness Constraint: encourages labels to be smooth on the manifold
- Fitting Constraint: encourages predictions to be consistent with initial labels

Label Propagation: Optimization Perspective

- Derive the closed-form solution

Notation: $\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_N \end{bmatrix}$

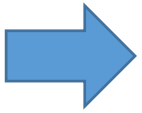
$$c = \frac{1}{2}\sum_i \sum_j w_{ij} \left\| \frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}} \right\|_2^2 + \beta \sum_i \left\| \mathbf{z}_i - \mathbf{y}_i \right\|_2^2$$

$$= \frac{1}{2}\sum_i \sum_j \left(w_{ij} \frac{\mathbf{z}_i^T \mathbf{z}_i}{d_i} + w_{ij} \frac{\mathbf{z}_j^T \mathbf{z}_j}{d_j} - 2w_{ij} \frac{\mathbf{z}_i^T \mathbf{z}_j}{\sqrt{d_i d_j}} \right) + \beta \sum_i \left(\mathbf{z}_i^T \mathbf{z}_i - 2\mathbf{z}_i^T \mathbf{y}_i \right) + C$$

$$= \sum_i \sum_j w_{ij} \frac{\mathbf{z}_i^T \mathbf{z}_i}{d_i} - \sum_i \sum_j w_{ij} \frac{\mathbf{z}_i^T \mathbf{z}_j}{\sqrt{d_i d_j}} + \beta \sum_i \left(\mathbf{z}_i^T \mathbf{z}_i - 2\mathbf{z}_i^T \mathbf{y}_i \right) + C$$

$$= \sum_i \mathbf{z}_i^T \mathbf{z}_i - \sum_i \sum_j \tilde{w}_{ij} \mathbf{z}_i^T \mathbf{z}_j + \beta \sum_i \left(\mathbf{z}_i^T \mathbf{z}_i - 2\mathbf{z}_i^T \mathbf{y}_i \right) + C$$

$$= tr\left(\mathbf{Z}^T \mathbf{Z}\right) - tr\left(\tilde{\mathbf{W}}^T \mathbf{Z}^T \mathbf{Z}\right) + \beta tr\left(\mathbf{Z}^T \mathbf{Z}\right) - \beta tr\left(\mathbf{Z}^T \mathbf{Y}\right) + C$$

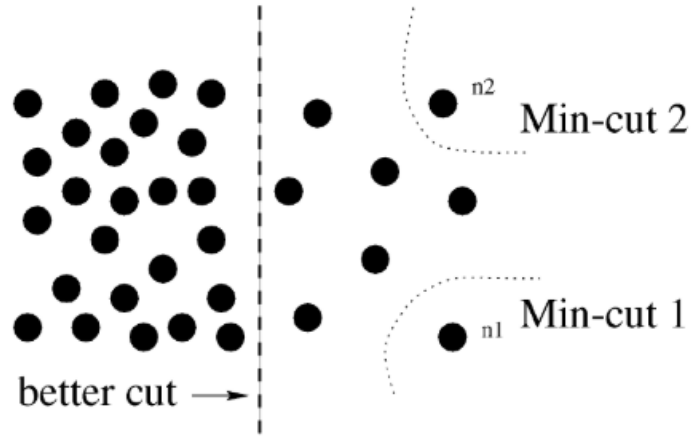


$$\frac{\partial c}{\partial \mathbf{Z}} = \mathbf{Z} - \tilde{\mathbf{W}}\mathbf{Z} + \beta\mathbf{Z} - \beta\mathbf{Y} = 0$$

$$\Rightarrow \left(\mathbf{I} - \frac{1}{1+\beta} \tilde{\mathbf{W}} \right) \mathbf{Z} = \frac{\beta}{1+\beta} \mathbf{Y}$$

$$\Rightarrow \mathbf{Z} = \beta\alpha \left(\mathbf{I} - \alpha\tilde{\mathbf{W}} \right)^{-1} \mathbf{Y} \quad s.t. \alpha = \frac{1}{1+\beta}$$

Relation to Spectral Clustering



- Normalized Cut (Ncut)[1]

$$Ncut\left(C_1,\cdots,C_K\right)=\sum_{k=1\cdots K}\frac{cut\left(C_k,\bar{C}_k\right)}{vol\left(C_k\right)}=\frac{1}{2}\sum_{k=1\cdots K}\frac{W\left(C_k,\bar{C}_k\right)}{vol\left(C_k\right)}$$

Partition all nodes into K clusters

$$C=C_1\bigcup C_2,\cdots,\bigcup C_K$$

Denote $f_i\in\left\{0,1\right\}^K$ as the partition index for node i

$$W\left(C_k,\bar{C}_k\right)=\sum_{i\in C_k}\sum_{j\in\bar{C}_k}w_{ij}\left\|f_i-f_j\right\|_2^2=\sum_{i\in C_k}\sum_{j\in C}w_{ij}\left\|f_i-f_j\right\|_2^2$$

$$vol\left(C_k\right)=\sum_{i\in C_k}\sum_{j\in C}w_{ij}=\sum_{i\in C_k}d_i$$

[1] Shi, J. and Malik, J., *Normalized cuts and image segmentation*, TPAMI 2001

Relation to Spectral Clustering - NCut

$$Ncut(C_1,\cdots,C_K)=\frac{1}{2}\sum_{k=1\cdots K}\frac{\sum_{i\in C_k}\sum_{j\in C}w_{ij}\left\|\mathbf{f}_i-\mathbf{f}_j\right\|_2^2}{\sum_{i\in C_k}d_i}$$

$$=\frac{1}{2}\sum_{k=1\cdots K}\sum_{i\in C_k}\sum_{j\in C}w_{ij}\left\|\tilde{\mathbf{f}}_i-\tilde{\mathbf{f}}_j\right\|_2^2=\frac{1}{2}\sum_{i\in C}\sum_{j\in C}w_{ij}\left\|\tilde{\mathbf{f}}_i-\tilde{\mathbf{f}}_j\right\|_2^2$$

$$=\frac{1}{2}\sum_{i\in C}\sum_{j\in C}w_{ij}\left\|\frac{\mathbf{z}_i}{\sqrt{d_i}}-\frac{\mathbf{z}_j}{\sqrt{d_j}}\right\|_2^2$$

$$\tilde{\mathbf{f}}_i=\frac{\mathbf{f}_i}{\sqrt{\sum_{i\in C_k}d_i}}\quad s.t.\quad i\in C_k$$

$$\tilde{\mathbf{F}}=\left[\underbrace{\tilde{\mathbf{f}}_1\cdots\tilde{\mathbf{f}}_{|C_1|}}_{|C_1|},\underbrace{\tilde{\mathbf{f}}_{|C_1|+1}\cdots\tilde{\mathbf{f}}_{|C_1|+|C_2|}}_{|C_2|},\cdots\underbrace{\tilde{\mathbf{f}}_{|C_1|+\cdots|C_{K-1}|}\cdots\tilde{\mathbf{f}}_N}_{|C_K|}\right]^T\qquad \tilde{\mathbf{F}}^T\mathbf{D}\tilde{\mathbf{F}}=\mathbf{I}$$

Let $\mathbf{Z}=\mathbf{D}^{0.5}\tilde{\mathbf{F}}$ We have $\mathbf{Z}^T\mathbf{Z}=\mathbf{I}$ and $\tilde{\mathbf{f}}_i=\frac{\mathbf{z}_i}{\sqrt{d_i}}$

The Final Objective

$$\min_{\{\mathbf{f}_i\}}Ncut(C_1,\cdots,C_K)$$

$$\Rightarrow\min_{\{\mathbf{z}_i\}}\frac{1}{2}\sum_{i\in C}\sum_{j\in C}w_{ij}\left\|\frac{\mathbf{z}_i}{\sqrt{d_i}}-\frac{\mathbf{z}_j}{\sqrt{d_j}}\right\|_2^2\quad s.t.\quad\mathbf{Z}^T\mathbf{Z}=\mathbf{I}$$

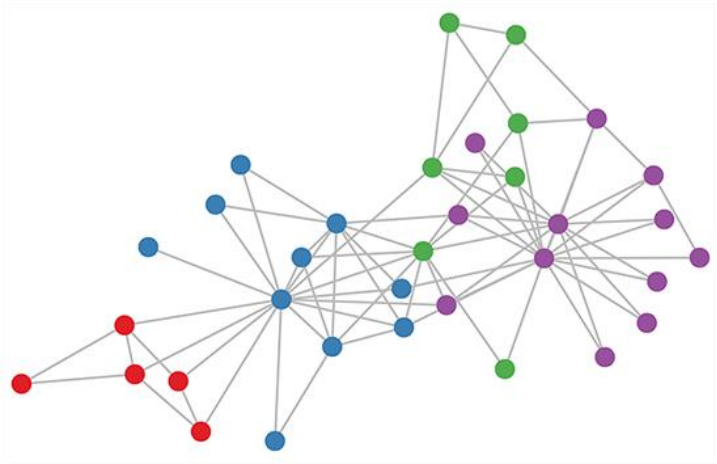
Solve by relaxing $\mathbf{Z}\in\mathbf{R}^{N\times K}$

Relation to GCN

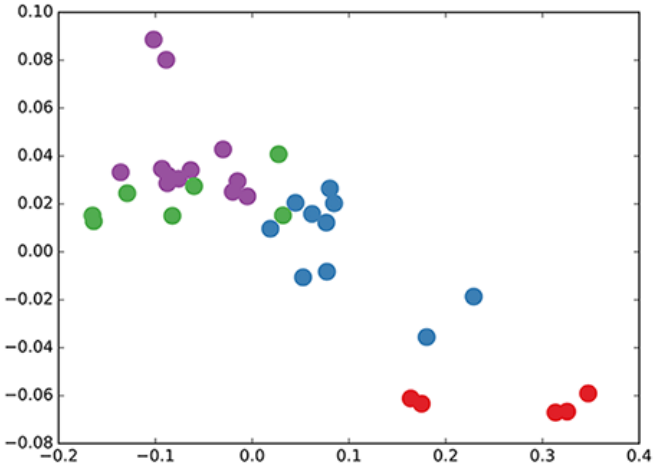
- The Propagation Rule for GCN [1]

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-0.5}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-0.5}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right) \qquad \mathbf{D} = diag\left(\sum_j \tilde{a}_{ij}\right), \quad \tilde{\mathbf{A}} = \mathbf{I} + \mathbf{A}$$

- Explain the Success of Random Initialization [1]



Random **W**
Identity $\mathbf{H}^0$



[1] Kipf, T. and Welling, M., *Semi-Supervised Classification with Graph Convolutional Networks*, ICLR 2017

Efficient Computation

- Large-Scale Label Propagation

$$\mathbf{Z} = \left(\mathbf{I} - \alpha\tilde{\mathbf{W}}\right)^{-1}\mathbf{Y}$$ The inverse is up to $O(n^3)$

- Iterative Method for $\left(\mathbf{I} - \alpha\tilde{\mathbf{W}}\right)\mathbf{Z} = \mathbf{Y}$
- Denote $\mathbf{A} = \mathbf{I} - \alpha\tilde{\mathbf{W}}, \quad \mathbf{b} = \mathbf{y}_i$
- Solve the unconstraint quadratic problem $\min_{\mathbf{z}_i} \mathbf{z}_i^T\mathbf{A}\mathbf{z}_i - \mathbf{b}\mathbf{z}_i$
- Unique solution exists $\mathbf{A} \succ 0$
- Gradient Descent $\mathbf{z}_{(t+1)} = \mathbf{z}_{(t)} - \alpha_{(t)}\mathbf{d}_{(t)}, \quad s.t.\mathbf{d}_{(t)} = \mathbf{A}\mathbf{z}_{(t)} - \mathbf{b}$
- Conjugate Descent [1]: How to find $\alpha_{(t)}$ and $\mathbf{d}_{(t)}$ to speed up convergence

[1] Shewchuk, Jonathan Richard. "An introduction to the conjugate gradient method without the agonizing pain." (1994): 1.

Self-Supervised Training

- Calculate Weight for Each Sample

$$\omega_i = 1 - \frac{H\left(\hat{\mathbf{z}}_i\right)}{\log\left(C\right)} \qquad \hat{z}_{ij} = z_{ij} \Big/ \sum_k z_{ik}$$

- Calculate Weight for Each Class

$$\zeta_j = \left(\left|\mathbf{L}_j\right| + \left|\mathbf{U}_j\right|\right)^{-1}$$

- The Final Loss

$$Lw\left(\mathbf{X}, \mathbf{Y}_l, \mathbf{Y}_u; \boldsymbol{\theta}\right) = \sum_{i=1\cdots l} \zeta_{y_i} l_s\left(f\left(\mathbf{x}_i; \boldsymbol{\theta}\right), \mathbf{y}_i\right) + \sum_{i=l+1\cdots n} \omega_i \zeta_{\hat{y}_i} l_s\left(f\left(\mathbf{x}_i; \boldsymbol{\theta}\right), \hat{\mathbf{y}}_i\right)$$

Summary of Algorithm

Algorithm 1 Label propagation for deep SSL

procedure LPDSSL(Training examples X, labels Y_L)
  $\theta \leftarrow$ initialize randomly
  for epoch $\in [1, \ldots, T]$ **do**
    $\theta \leftarrow \textsc{Optimize}(L_s(X_L, Y_L; \theta))$ ▷ mini-batch optimization
  end for
  for epoch $\in [1, \ldots, T']$ **do**
    for $i \in \{1, \ldots, n\}$ **do** $\mathbf{v}_i \leftarrow \phi_\theta(x_i)$ ▷ extract descriptors
    for $(i, j) \in \{1, \ldots, n\}^2$ **do** $a_{ij} \leftarrow$ affinity values (9)
    $W \leftarrow A + A^\top$ ▷ symmetric affinity
    $\mathcal{W} \leftarrow D^{-1/2} W D^{-1/2}$ ▷ symmetrically normalized affinity
    $Z \leftarrow$ solve (10) with CG ▷ diffusion
    for $(i, j) \in U \times C$ **do** $\hat{z}_{ij} \leftarrow z_{ij} / \sum_k z_{ik}$ ▷ normalize Z
    for $i \in U$ **do** $\hat{y}_i \leftarrow \arg\max_j \hat{z}_{ij}$ ▷ pseudo-label
    for $i \in U$ **do** $\omega_i \leftarrow$ certainty of $\hat{y}_i$ (11) ▷ pseudo-label weight
    for $j \in C$ **do** $\zeta_j \leftarrow (|L_j| + |U_j|)^{-1}$ ▷ class weight/balancing
    $\theta \leftarrow \textsc{Optimize}(L_w(X, Y_L, \hat{Y}_U; \theta))$ ▷ mini-batch optimization
  end for
end procedure

Experiments

- Dataset

Dataset	#Classes (tr/te)	#Samples (tr/te)	#Labelled Samples per Class	Image Res.	Network
CIFAR-10	10/10	50k/10k	50/100/200/400	32*32	’13layer’ CNN
CIFAR-100	100/100	50k/10k	40/100	32*32	’13layer’ CNN
Mini-ImageNet	100/100	50k/10k	40/100	84*84	Resnet-18

- Some Examples

Experiment - Comparisons

- Virtual Adversarial Training [1]
- Adversarial Training with full supervision [2]

$$\min_{\boldsymbol{\theta}} \sum_{i} D\left(f\left(\mathbf{X}_i + \hat{\eta}; \boldsymbol{\theta}\right), \mathbf{y}_i \right)$$

$$where \quad \hat{\eta} = \arg\max_{\eta; \|\eta\| \le \varepsilon} D\left(f\left(\mathbf{X}_i + \eta; \boldsymbol{\theta}\right), \mathbf{y}_i \right)$$

$D(\cdot,\cdot)$ is a divergence and $\mathbf{y}_i$ are known labels

- Virtual Adversarial Training with semi-supervision [1]

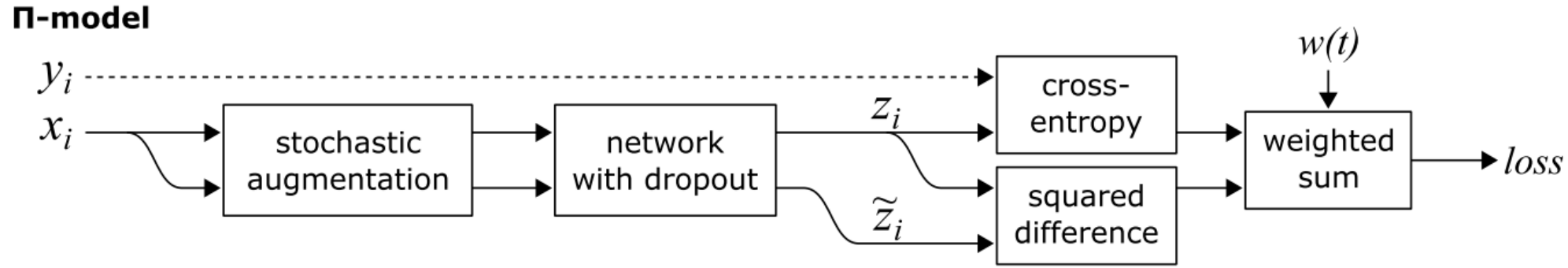
$$\min_{\boldsymbol{\theta}} \sum_{i \in \mathbf{L}} D\left(f\left(\mathbf{X}_i + \hat{\eta}; \boldsymbol{\theta}\right), \mathbf{y}_i \right) + \sum_{i \in \mathbf{U}} D\left(f\left(\mathbf{X}_i + \hat{\eta}; \boldsymbol{\theta}^{(t)}\right), f\left(\mathbf{X}_i; \boldsymbol{\theta}^{(t-1)}\right) \right)$$

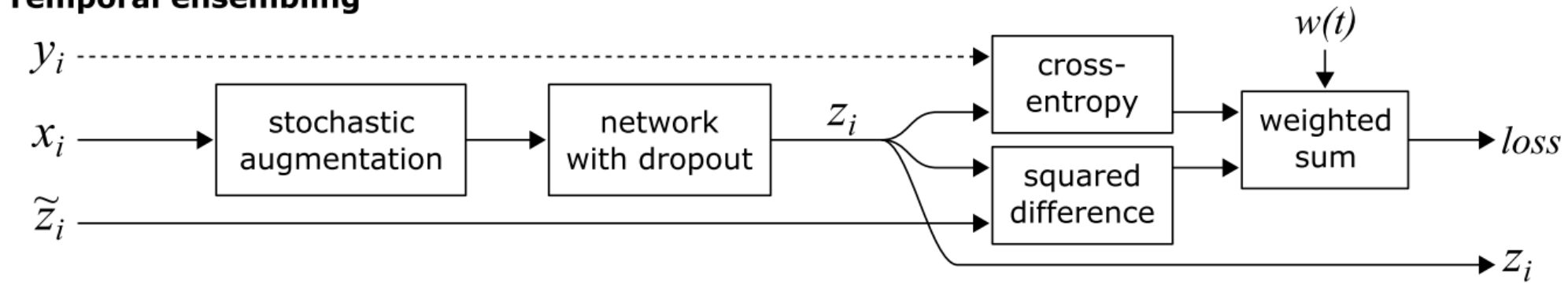
$$where \quad \hat{\eta} = \begin{cases} \arg\max_{\eta; \|\eta\| \le \varepsilon} D\left(f\left(\mathbf{X}_i + \eta; \boldsymbol{\theta}\right), f\left(\mathbf{X}_i; \boldsymbol{\theta}^{(t-1)}\right) \right) & i \in \mathbf{U} \\ \arg\max_{\eta; \|\eta\| \le \varepsilon} D\left(f\left(\mathbf{X}_i + \eta; \boldsymbol{\theta}\right), \mathbf{y}_i \right) & i \in \mathbf{L} \end{cases}$$

[1] Miyato, Takeru, et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning." *TPAMI. 2018*
[2] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explain- ing and harnessing adversarial examples.” *ICLR*, 2015.

Experiment - Comparisons

- Temporal Ensemble & Π-model [1]



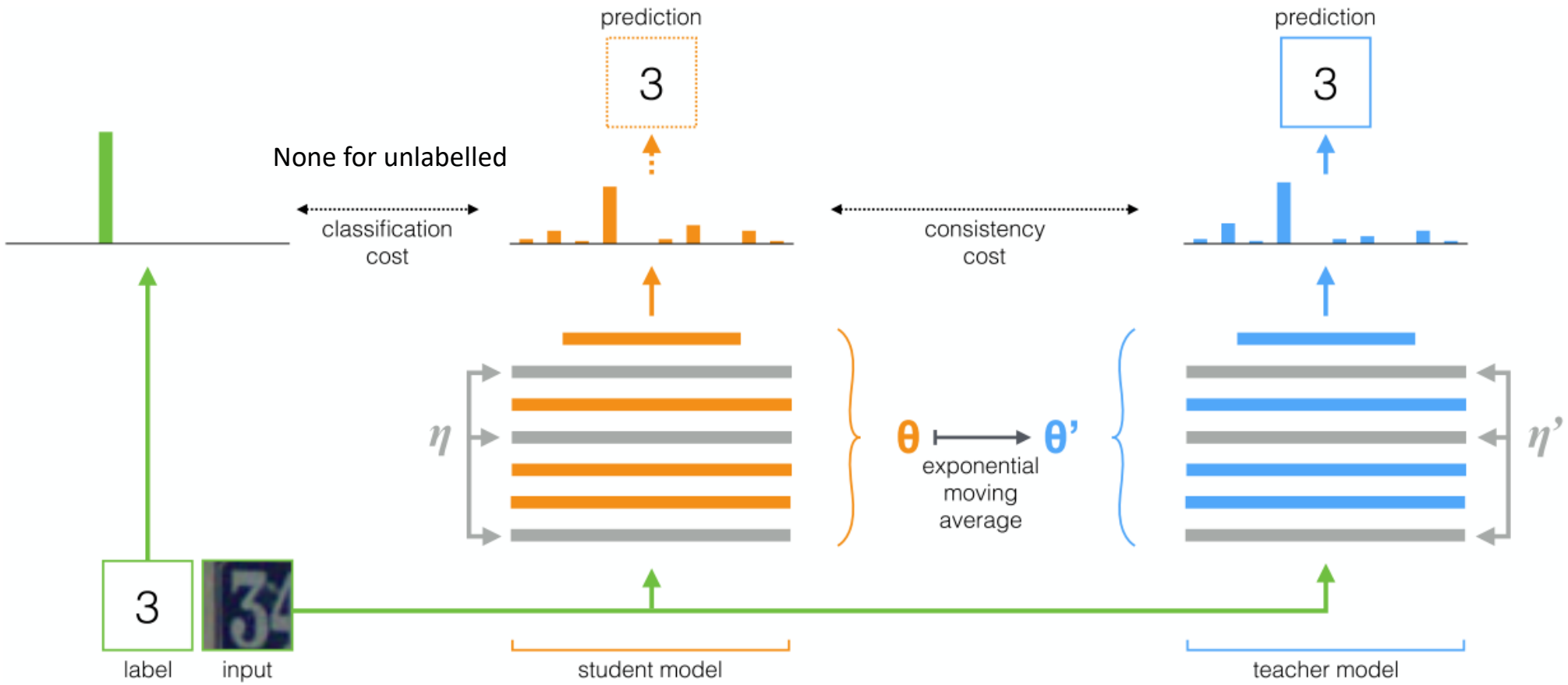


Moving Average $\tilde{z}_i^{(t)} = \alpha \tilde{z}_i^{(t-1)} + \frac{(1-\alpha) z_i^{(t-1)}}{1-\alpha^t}$

[1] Laine, Samuli, and Timo Aila. "Temporal Ensembling for Semi-Supervised Learning." *ICLR.* 2016.

Experiment - Comparisons

- Mean Teacher (MT) [1]:



Random Noise for Input

η and η'

Moving Average for TeacherNet

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$$

Total Loss

$$\min_{\theta} \left\| f(\mathbf{X}_i + \eta; \theta) - f(\mathbf{X}_i + \eta'; \theta') \right\|_2^2 + \left\| f(\mathbf{X}_i + \eta; \theta) - \mathbf{y}_i \right\|_2^2$$

[1] Tarvainen, Antti, and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results." *NIPS*. 2017.

Experiment - Comparisons

- TDCNN [2]:

Iterative Training

Training on Labelled Samples

$$\min_{\boldsymbol{\theta}^{(0)}} \sum_{i \in \mathbf{L}} CE\left(f\left(\mathbf{X}_i; \boldsymbol{\theta}^{(0)}\right), \mathbf{y}_i \right) \Rightarrow \min_{\boldsymbol{\theta}_t} \sum_{i \in \mathbf{U}} r_i CE\left(f\left(\mathbf{X}_i; \boldsymbol{\theta}^{(t)}\right), f\left(\mathbf{X}_i; \boldsymbol{\theta}^{(t-1)}\right)\right) + \sum_{i \in \mathbf{L}} CE\left(f\left(\mathbf{X}_i; \boldsymbol{\theta}^{(t)}\right), \mathbf{y}_i \right) + L_{reg}$$

Per-sample weight depends on density

$$d_i = \sum_{j \in NNk(f_i)} \left\| f_i - f_j \right\|_2^2$$

$$r_i = 1 - \frac{d_i}{d_{\max}}$$

Min-Max Feature Regularization

- Samples of same class to be close
- Samples of different classes to be faraway, above a threshold

$$L_{reg} = r_i r_j \left(\mathbf{1}\left(y_i = y_j\right) \left\| f_i - f_j \right\|_2^2 - \left(1 - \mathbf{1}\left(y_i = y_j\right)\right) \min\left(0, \left\| f_i - f_j \right\|_2^2 - \tau\right)\right)$$

[1] Shi, Weiwei, et al. "Transductive semi-supervised deep learning using min-max features." *ECCV*. 2018.

Experiment - Quantitative Evaluation on All 3 Datasets

Dataset	CIFAR-100		Mini-ImageNet-*top1*		Mini-ImageNet-*top5*	
Nb. labeled images	4000	10000	4000	10000	4000	10000
Fully supervised	55.43 ± 0.11	40.67 ± 0.49	74.78 ± 0.33	60.25 ± 0.29	53.07 ± 0.68	38.28 ± 0.38
Ours	46.20 ± 0.76	38.43 ± 1.88	$\mathbf{70.29 \pm 0.81}$	57.58 ± 1.47	$\mathbf{47.58 \pm 0.94}$	36.14 ± 2.19
MT [38]	45.36 ± 0.49	36.08 ± 0.51	72.51 ± 0.22	57.55 ± 1.11	49.35 ± 0.22	32.51 ± 1.31
MT + Ours	$\mathbf{43.73 \pm 0.20}$	$\mathbf{35.92 \pm 0.47}$	72.78 ± 0.15	$\mathbf{57.35 \pm 1.66}$	50.52 ± 0.39	$\mathbf{31.99 \pm 0.55}$

Table 3. Performance comparison on CIFAR-100 and Mini-ImageNet with 4k and 10k labeled images. Error rate is reported. "13-layer" network is used for CIFAR-100 and Resnet-18 is used for Mini-ImageNet. All methods are reproduced by us.

Experiment – Comparison on CIFAR-10

Dataset	CIFAR-10			
Nb. labeled images	500	1000	2000	4000
Fully supervised	49.08 ± 0.83	40.03 ± 1.11	29.58 ± 0.93	21.63 ± 0.38
TDCNN [36]†	-	32.67 ± 1.93	22.99 ± 0.79	16.17 ± 0.37
Network prediction (1) + weights	35.17 ± 2.46	23.79 ± 1.31	16.64 ± 0.48	13.21 ± 0.61
Ours: Diffusion prediction (7) + weights	32.40 ± 1.80	22.02 ± 0.88	15.66 ± 0.35	12.69 ± 0.29
VAT [26]†	-	-	-	11.36
Π model [23]†	-	-	-	12.36 ± 0.31
Temporal Ensemble [23]†	-	-	-	12.16 ± 0.24
MT [38]†	-	27.36 ± 1.30	15.73 ± 0.31	12.31 ± 0.28
MT [38]	27.45 ± 2.64	19.04 ± 0.51	14.35 ± 0.31	11.41 ± 0.25
MT + Ours	$\mathbf{24.02 \pm 2.44}$	$\mathbf{16.93 \pm 0.70}$	$\mathbf{13.22 \pm 0.29}$	$\mathbf{10.61 \pm 0.28}$

Table 2. Comparison with the state of the art on CIFAR-10. Error rate is reported. "13-layer" network is used. The top part of the table corresponds to training with pseudo-labels, while the bottom part of the table includes methods that are complementary to ours, as shown by the combination of our method with MT. † denotes scores reported in prior work.

Experiments - Ablation Studies

Contribution of weighted learning

Pseudo-labeling	ω_i	ζ_j	CIFAR-10
Diffusion (7)			36.53 ± 1.42
		✓	36.17 ± 1.98
	✓		33.32 ± 1.53
	✓	✓	$\mathbf{32.40 \pm 1.80}$
GTG [8]	✓	✓	35.20 ± 2.23
Network (1)	✓	✓	35.17 ± 2.46

Table 1. Impact of weights ω_i, class weights ζ_j, and pseudo-labeling by diffusion prediction (7) or network prediction (1). Error rate is reported on CIFAR-10 with 500 labels.

Contribution of Label Propagation

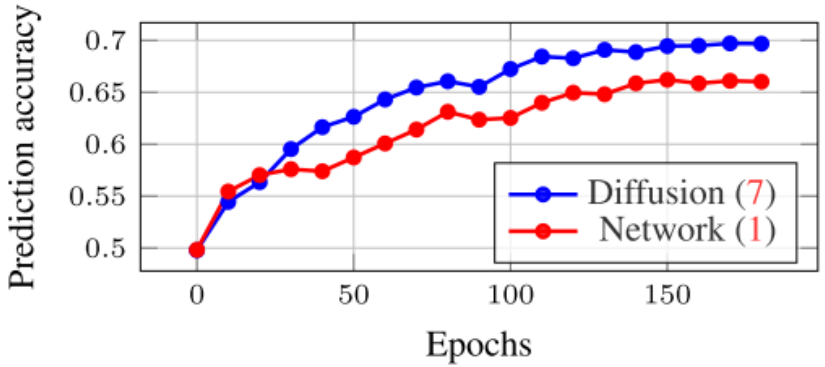


Figure 4. Accuracy of predicted pseudo-labels according to ground-truth on CIFAR-10 with 500 labeled images. Diffusion predictions (7) are compared against network predictions (1).

Evolution of Weight

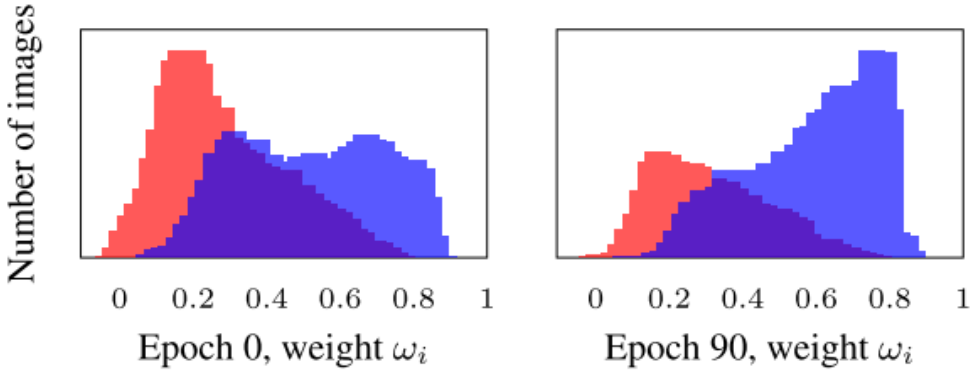


Figure 5. Distribution of weights ω_i for unlabeled images at epoch 0 (left) and epoch 90 (right) during the training of CIFAR-10 with 500 labels. Correct pseudo-labels according to ground-truth are shown in blue and incorrect in red.

Varying # Labelled Data

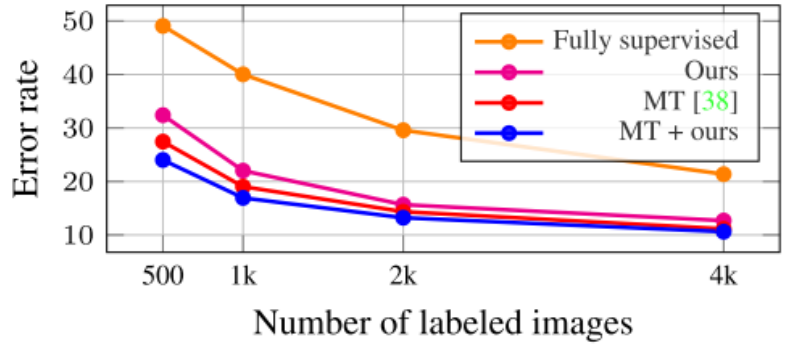
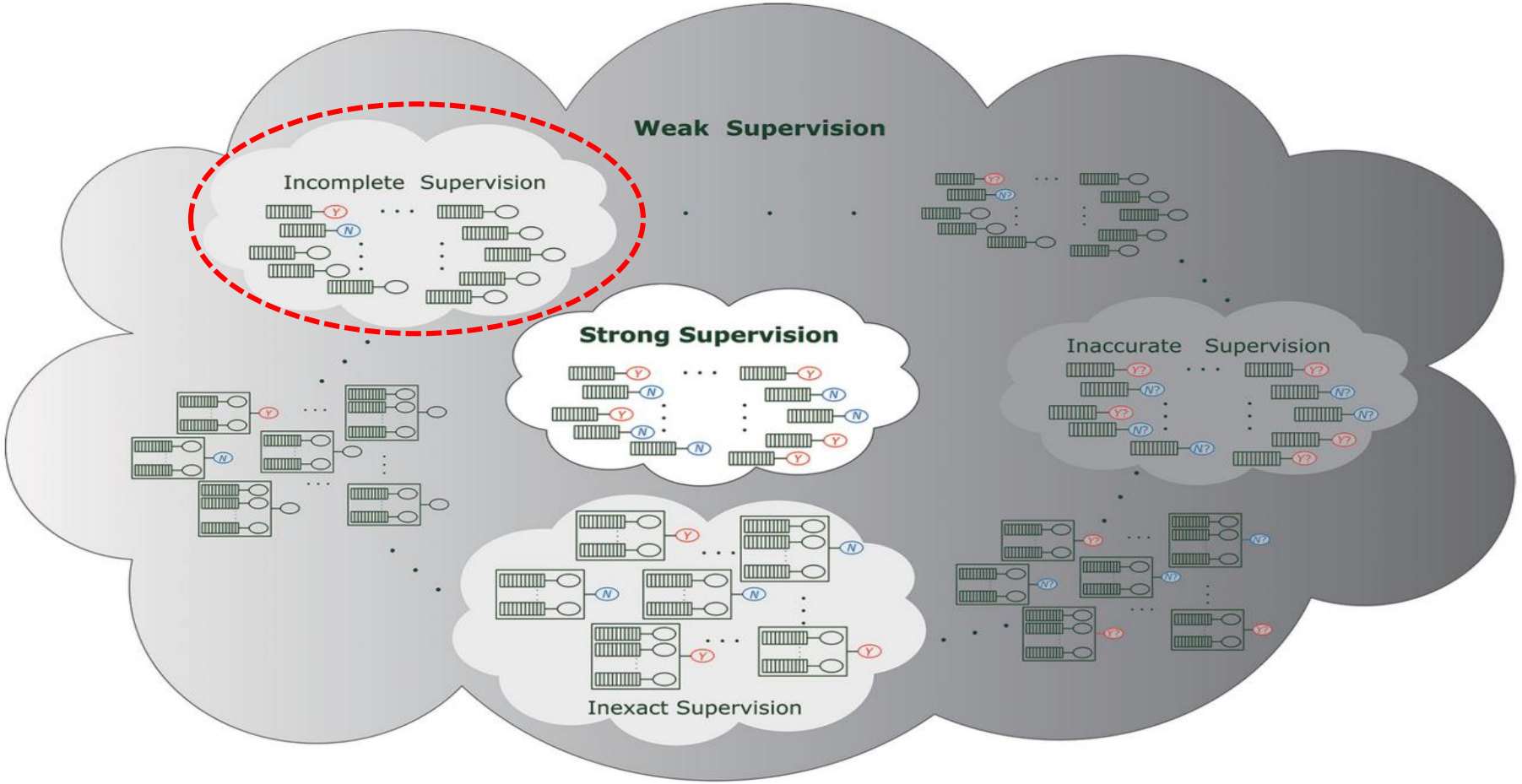


Figure 6. Error rate versus number of labeled images on CIFAR-10 using different methods.

Future Extension

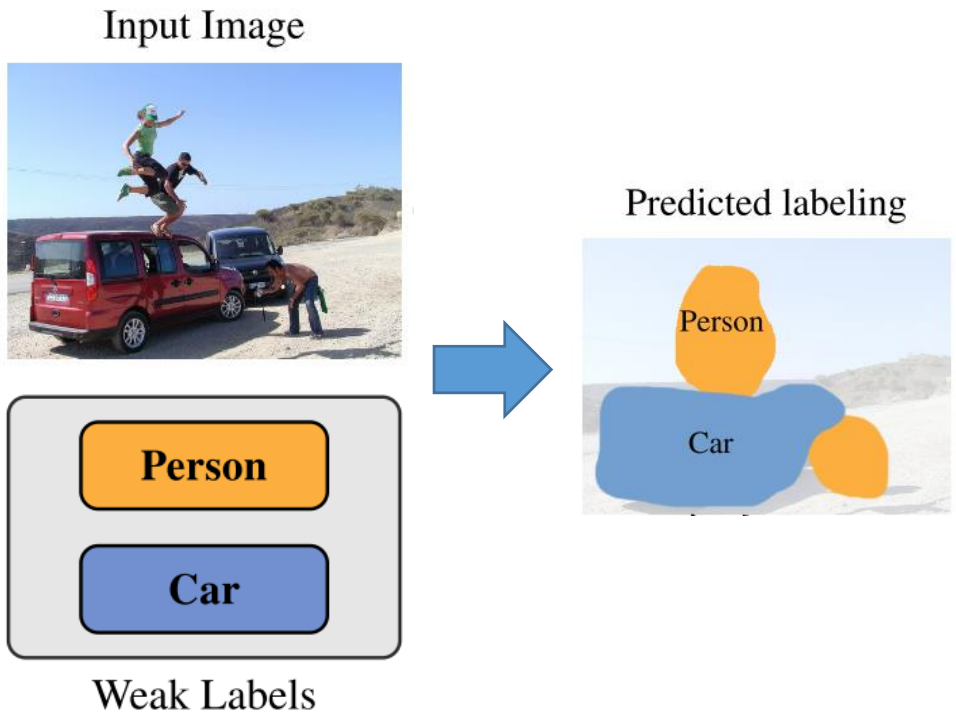
- Semi-Supervised Learning in the Spectrum of Weak Supervision [1]

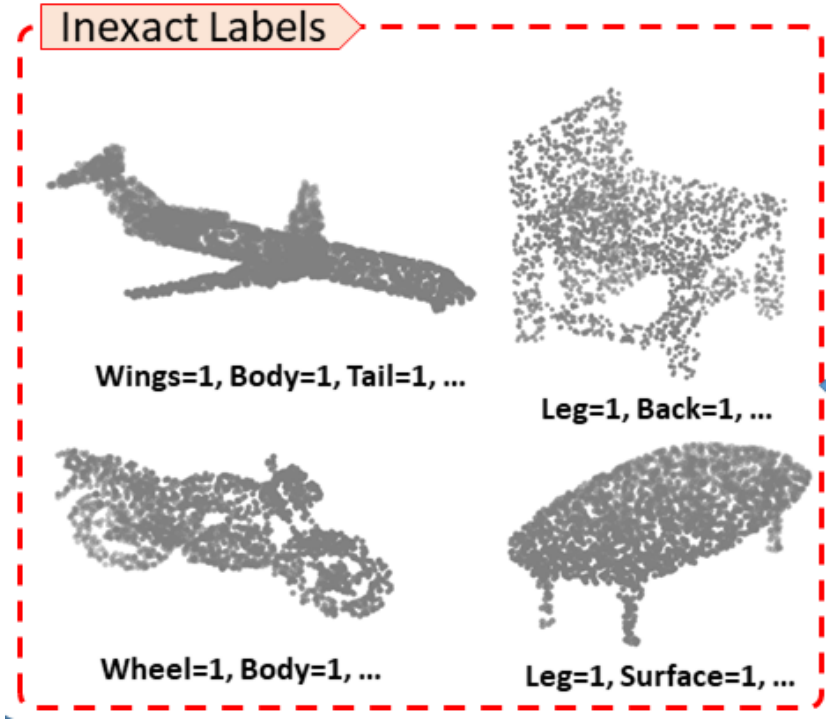


[1] Zhou, Zhi-Hua. "A brief introduction to weakly supervised learning." *National Science Review* 5.1 (2017): 44-53.

Future Extension

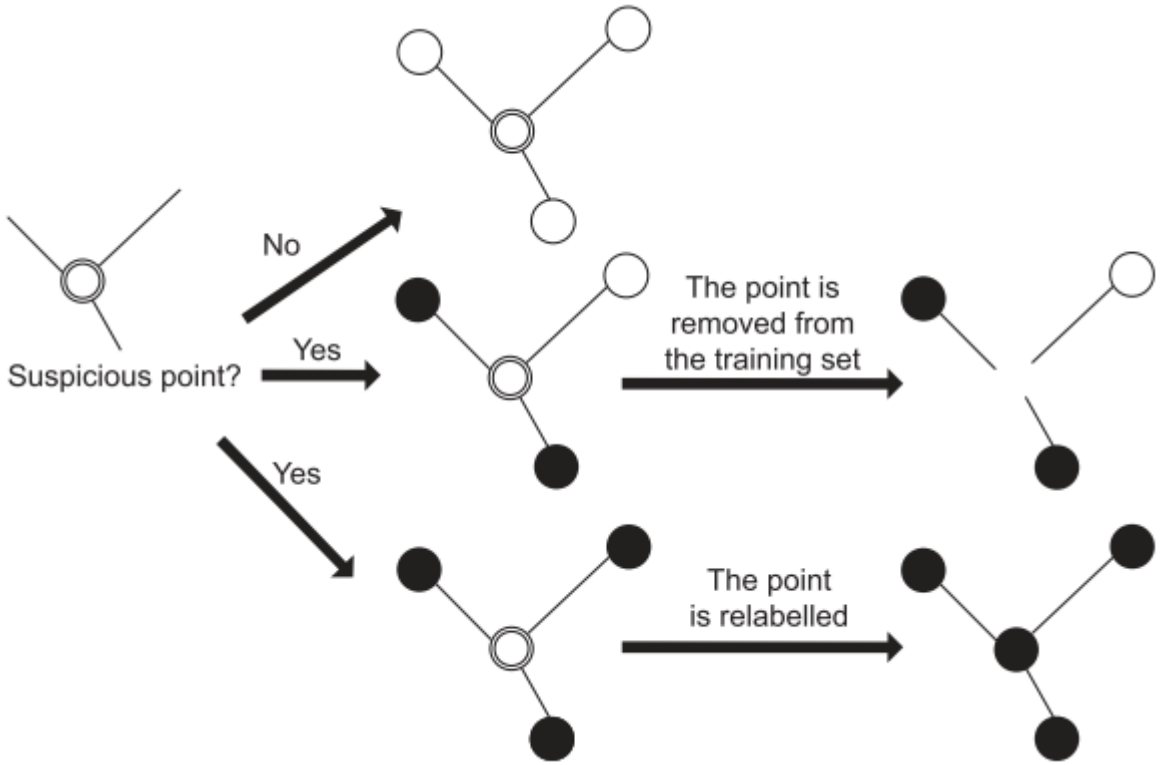
- Inexact Supervision (Weakly Supervised Learning)
- Full annotation is too expensive and is not scalable, e.g. image segmentation & point cloud segmentation





Future Extension

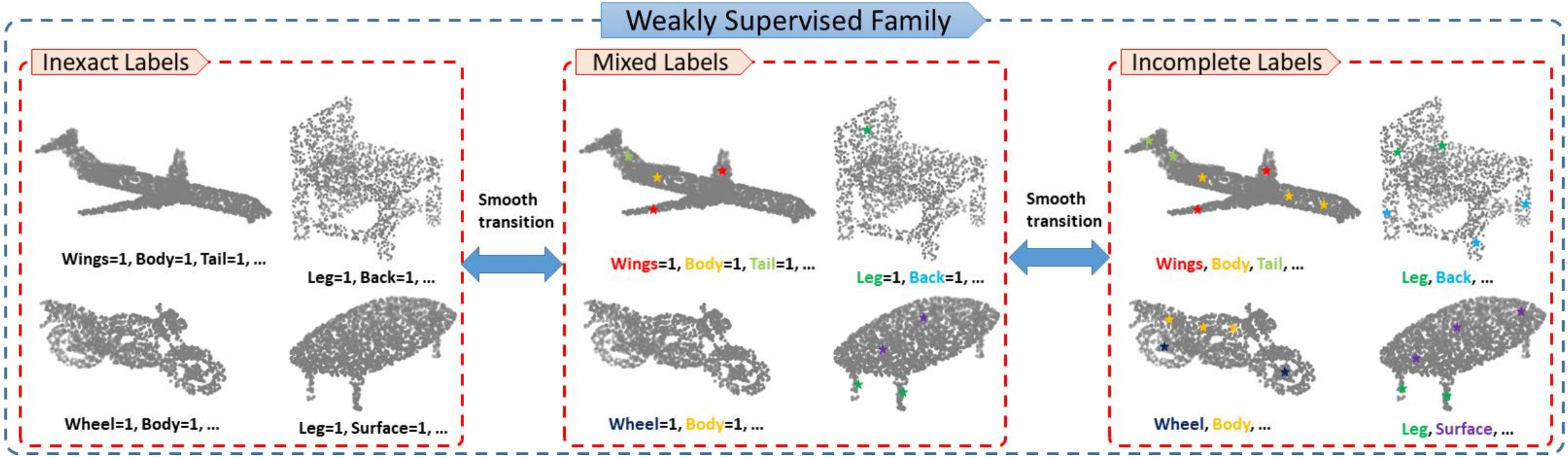
- Inaccurate Supervision (Some Labels are Wrong)
- Annotation is challenging or annotators are intentional, e.g. per-pixel/point annotation, robot annotation & adversarial annotations



[1] Zhou, Zhi-Hua. "A brief introduction to weakly supervised learning." *National Science Review* 5.1 (2017): 44-53.

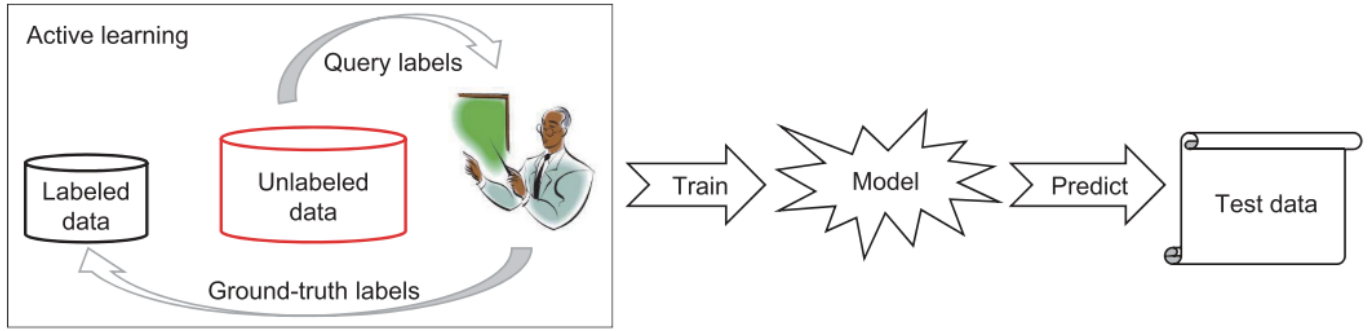
Future Extension

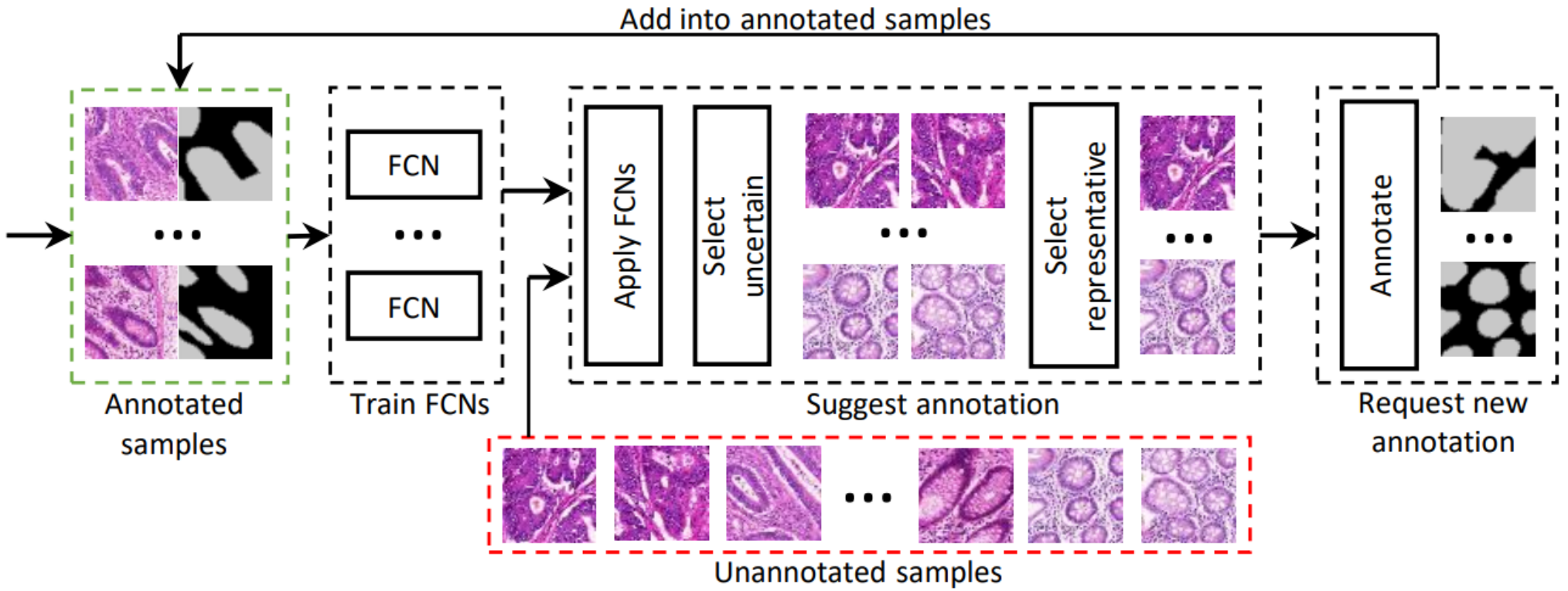
- At the Intersection of Incomplete and Inexact Labels



Future Extension

- Active Learning





[1] Yang, Lin, et al. "Suggestive annotation: A deep active learning framework for biomedical image segmentation." *International conference on medical image computing and computer-assisted intervention*. Springer, Cham, 2017.

Conclusion

- Exploiting unlabelled data for learning tasks is promising
- Relation to traditional graph theorem is revitalizing
- Various forms of weak supervision and/or less costly annotation could be explored