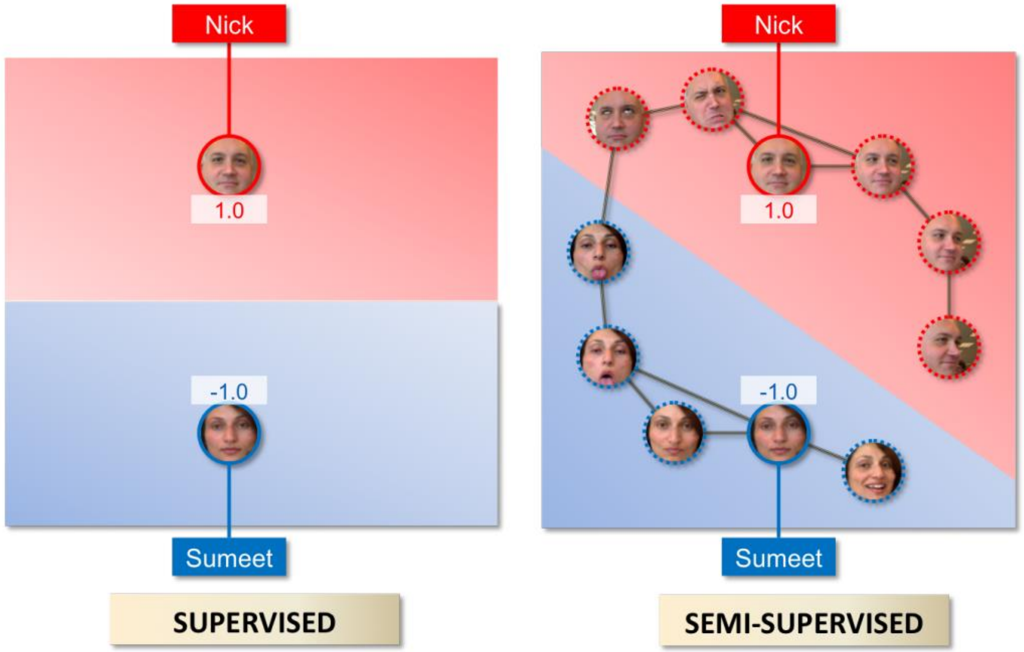
SSL Continue

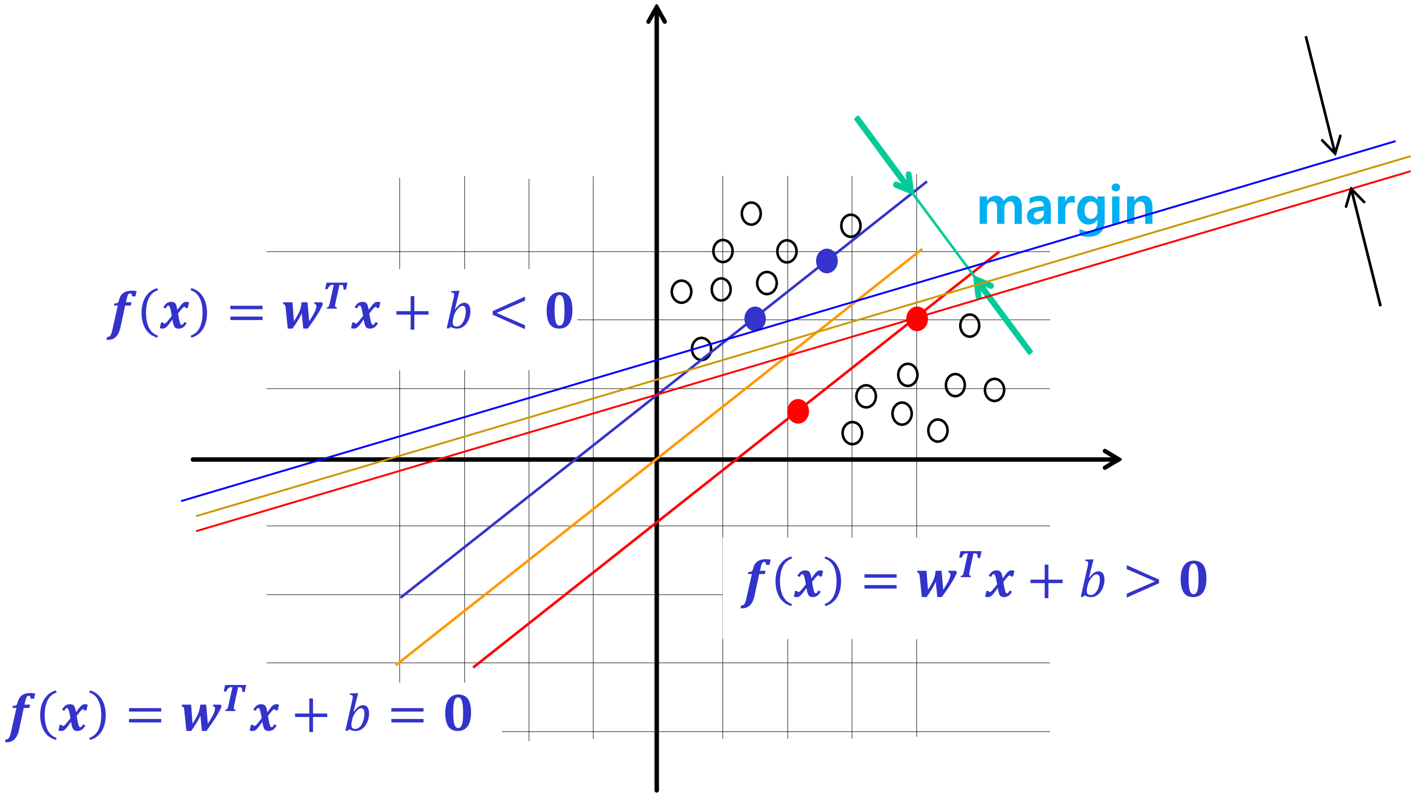
semi-supervised learning



Outline of Lecture (1)

- Graph Spectral Theory
  - Definition of Graph
  - Graph Laplacian
  - Laplacian Smoothing
- Graph Node Clustering
  - Minimum Graph Cut
  - Ratio Graph Cut
  - Normalized Graph Cut
- Manifold Learning
  - Spectral Analysis in Riemannian Manifolds
  - Dimension Reduction, Node Embedding
- Semi-supervised Learning (SSL)
  - Self-Training Methods
  - SSL with SVM
  - SSL with Graph using MinCut
  - SSL with Graph using Harmonic Functions
- Semi-supervised Learning (SSL) : conti.
  - SSL with Graph using Regularized Harmonic Functions
  - SSL with Graph using Soft Harmonic Functions
  - SSL with Graph using Manifold Regularization
  - SSL with Graph using Laplacian SVMs
  - SSL with Graph using Max-Margin Graph Cuts
  - Online SSL and SSL for large graph
- Graph Convolution Networks (GCN)
  - Graph Filtering in GCN
  - Graph Pooling in GCN
  - Spectral Filtering in GCN
  - Spatial Filtering in GCN
- Recent GCN papers

SSL: Classical SVM (Review)



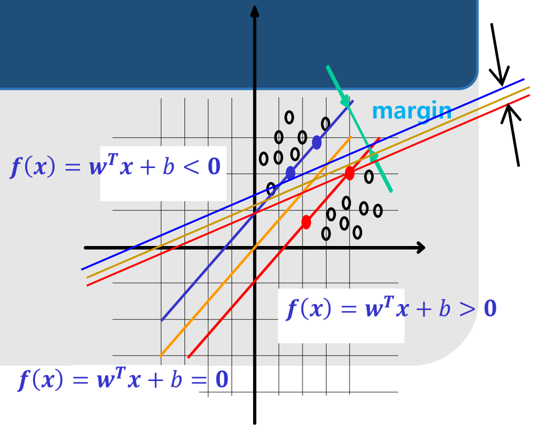
Maximal Margin Hyperplane = Optimal Hyperplane

SSL: Classical SVM (Review)

max-margin classification: separable case

$$\min_{\boldsymbol{w},b} \ \|\boldsymbol{w}\|^2$$

$$\boldsymbol{s.t.}\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \qquad \forall\, i = 1, \dots, n$$



Maximal Margin Hyperplane = Optimal Hyperplane

max-margin classification: non-separable case

$$\min_{\boldsymbol{w},b} \ \lambda\|\boldsymbol{w}\|^2 + \sum_i \xi_i$$

$$\boldsymbol{s.t.}\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \qquad \forall\, i = 1, \dots, n$$

$$\xi_i \geq 0, \quad \forall\, i = 1, \dots, n$$

SSL: Classical SVM (Review)

max-margin classification: non-separable case

$$\min_{\boldsymbol{w},b} \ \lambda\|\boldsymbol{w}\|^2 + \sum_i \xi_i$$
$$\boldsymbol{s.t.} \ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \qquad \forall\, i = 1, \dots, n$$
$$\xi_i \geq 0, \quad \forall\, i = 1, \dots, n$$

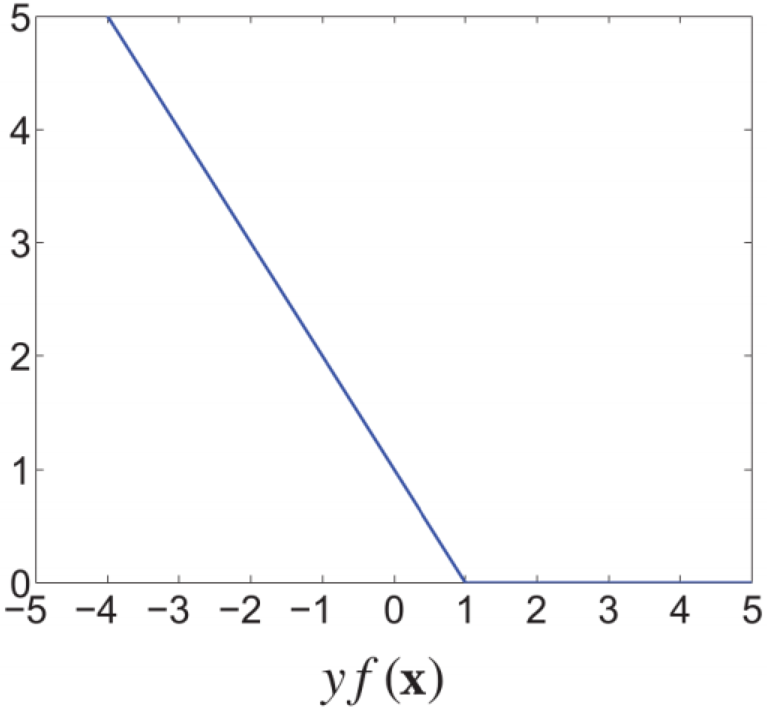
Unconstrained formulation using hinge loss:

$$\min_{\boldsymbol{w},b} \ \lambda\|\boldsymbol{w}\|^2 + \sum_i \max(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b), 0)$$

General formulation:

$$\min_{\boldsymbol{w},b} \ \lambda\Omega(\boldsymbol{f}(\boldsymbol{w}, b)) + \sum_i \Phi(\boldsymbol{x}_i, y_i, \boldsymbol{f}(\boldsymbol{w}, b; \boldsymbol{x}_i))$$

SSL: Classical SVM (Review)



hinge loss

$$\Phi(\boldsymbol{x}_i, y_i, \boldsymbol{f}(\boldsymbol{w}, b; \boldsymbol{x}_i)) = \max(1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b), 0)$$

SSL: SVM: Unlabeled Examples

Unconstrained formulation using hinge loss:

$$\min_{\boldsymbol{w},b} \ \lambda\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n_l} \max(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b), 0)$$

How to incorporate unlabeled examples?

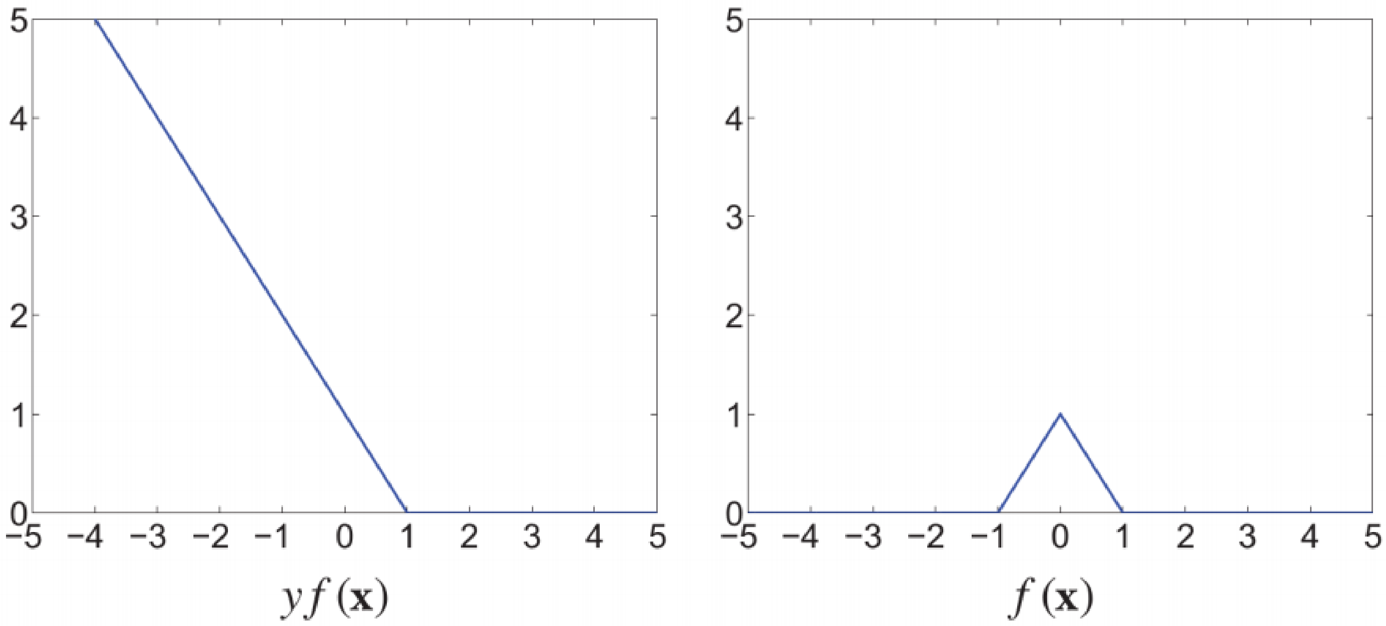
Prediction of f for (any) $\boldsymbol{x}$?

$$\hat{y}_i = sgn\big(f(\boldsymbol{x}_i)\big) = sgn(\boldsymbol{w}^T\boldsymbol{x}_i + b)$$

Use $\hat{y}_i$ instead of y_i

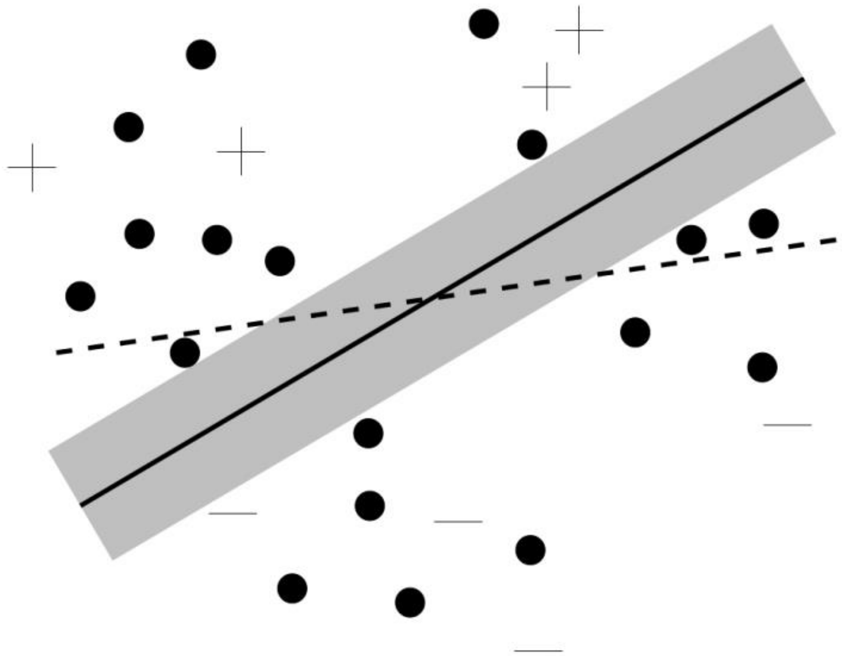
$$\begin{aligned}\Phi(\boldsymbol{x}_i, y_i, \boldsymbol{f}(\boldsymbol{w}, b; \boldsymbol{x}_i)) &= \max(1 - \hat{y}_i(\boldsymbol{w}^T\boldsymbol{x}_i + b), 0)\\ &= \max(1 - sgn(\boldsymbol{w}^T\boldsymbol{x}_i + b)(\boldsymbol{w}^T\boldsymbol{x}_i + b), 0)\\ &= \max(1 - |\boldsymbol{w}^T\boldsymbol{x}_i + b|, 0) \quad \Rightarrow \text{hat loss}\end{aligned}$$

SSL: SVM: Unlabeled Examples



(a) the hinge loss

(b) the hat loss



What is the difference in the objectives?

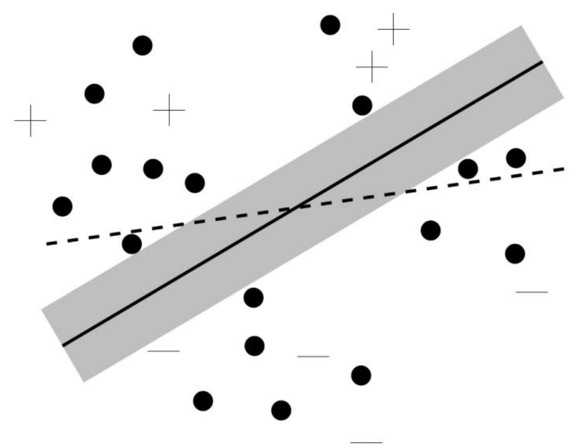
What does hinge loss penalize?

What does hat loss penalize?

SSL: SVM: Formulation

Formulation fo SSL via SVM

$$\min_{\boldsymbol{w},b} \sum_{i=1}^{n_l} \max(1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b), 0) + \lambda_1 \|\boldsymbol{w}\|^2$$
$$+\lambda_2 \sum_{i=n_l+1}^{n_l+n_u} \max(1 - |\boldsymbol{w}^T \boldsymbol{x}_i + b|, 0)$$

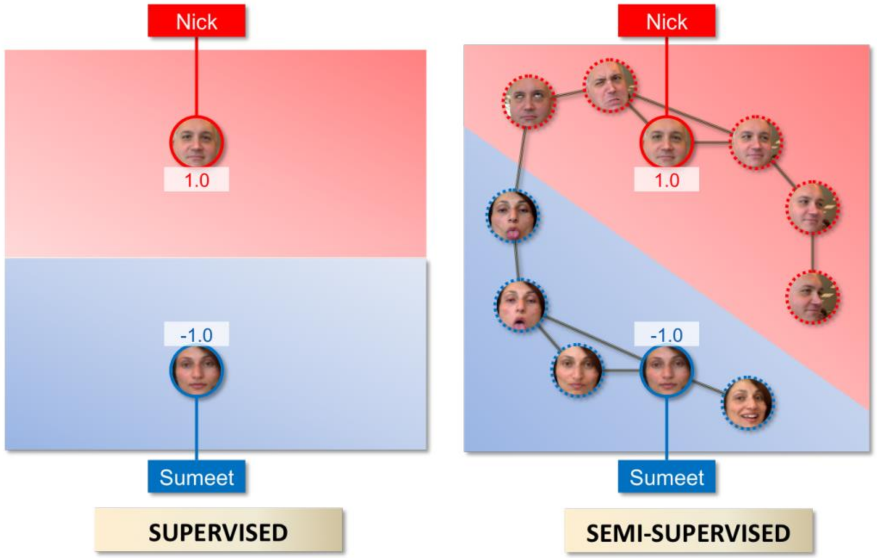


- Labelled data term works as a loss to learn the data
- Unlabeled data term works as a regularizer to reduce the effect of noisy data.
- The term $\|\boldsymbol{w}\|$ works as a regularizer for a large margin.

$SSL(\mathcal{G})$

semi-supervised learning

with graphs and harmonic functions

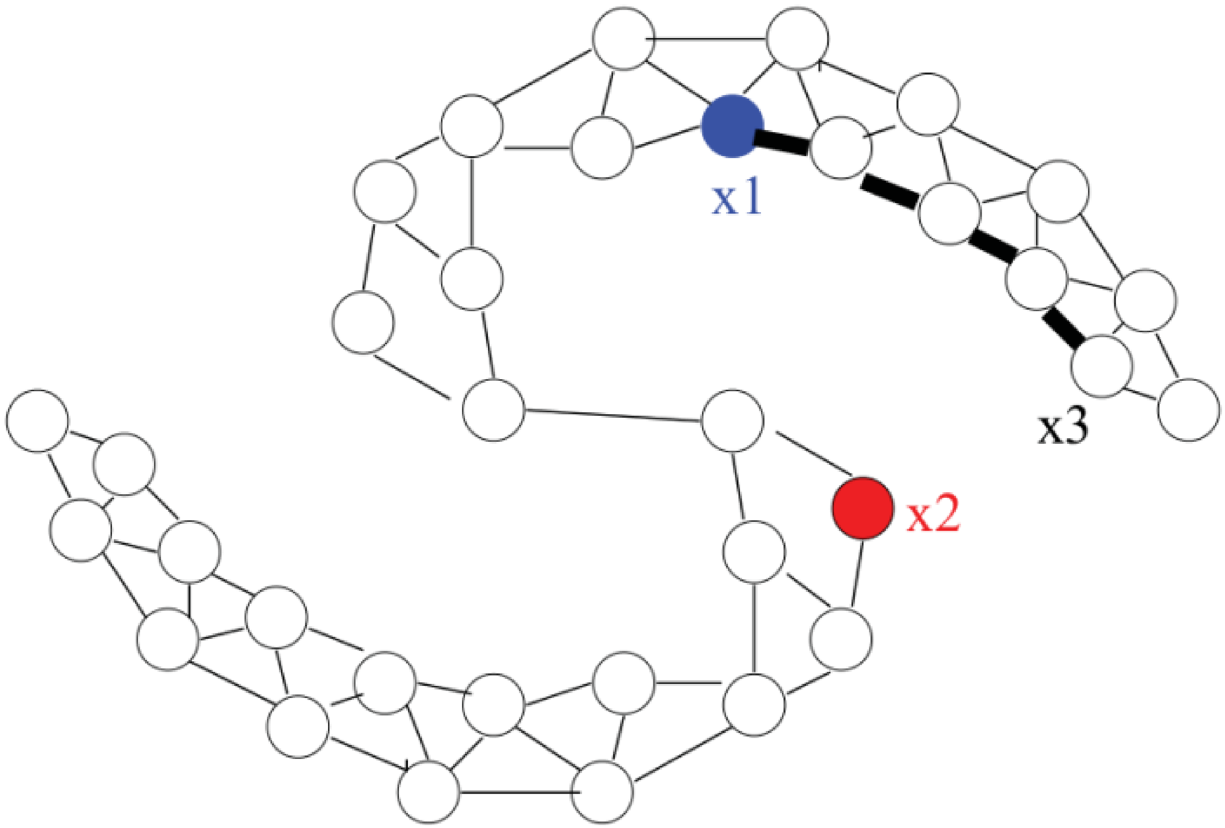


SSL with Graphs: Prehistory

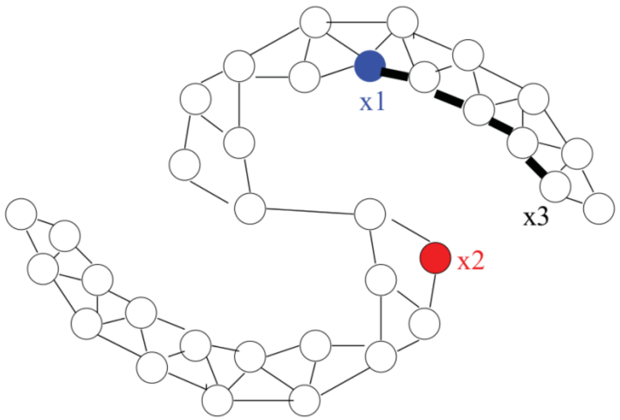
Blum/Chawla: Learning from Labeled and Unlabeled Data using Graph Mincuts

http://www.aladdin.cs.cmu.edu/papers/pdfs/y2001/mincut.pdf

Some insights from vision research in 1980s



SSL with Graphs: MinCut



MinCut SSL: an idea similar to MinCut clustering

Where is the link?

What is the formal statement? We look for $f(x) \in \{\pm 1\}$

$$cut = \sum_{i,j=1}^{n_l+n_u} w_{ij}\left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2 = \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f} = \Omega(f)$$

$\min_{f(\boldsymbol{x}_i);\, \boldsymbol{x}_i \in \boldsymbol{\mathcal{U}}} \Omega(f)$ ⟶ minimal smoothness for unsupervised clustering

What to do for semi-supervised learning?

SSL with Graphs: using $f(\boldsymbol{\theta}; \boldsymbol{x}_i)$

Inductive SSL with graph: using $f(\boldsymbol{\theta}; \boldsymbol{x}_i)$ classifier with parameters $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\boldsymbol{\theta}; \boldsymbol{x}_i) - f(\boldsymbol{\theta}; \boldsymbol{x}_j)\right)^2 + \gamma \sum_{i}^{n_l} (f(\boldsymbol{\theta}; \boldsymbol{x}_i) - y_i)^2$$

General Formulation

Regularization: Laplacian smoothing

$$\Omega\left(\{f(\boldsymbol{\theta}; \boldsymbol{x}_i)\}_{i=1}^{n_l+n_u}\right) = \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\boldsymbol{\theta}; \boldsymbol{x}_i) - f(\boldsymbol{\theta}; \boldsymbol{x}_j)\right)^2 = \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$$

Loss:

$$\Phi(f(\boldsymbol{\theta}; \boldsymbol{x}_i), y_i) = (f(\boldsymbol{\theta}; \boldsymbol{x}_i) - y_i)^2 \quad \forall\, i \in \{1, \ldots, n_l\}$$

SSL with Graphs: Harmonic Functions

Transductive **SSL with graph**: fixing $f(\boldsymbol{x}_i)$ for $i \in \mathcal{L}$

$$\min_{f \in \{\pm \mathbf{1}\}^{n_l+n_u}} \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2 + \infty \sum_{i}^{n_l} (f(\boldsymbol{x}_i) - y_i)^2$$

Solution:

An integer program: NP hard

Can we use eigenvectors? No. Why?

We need a better way to reflect the confidence.

SSL with Graphs: Harmonic Functions

Relaxation: Transductive SSL with graph: fixing $f(\boldsymbol{x}_i)$ for $i \in \mathcal{L}$

$$\min_{\boldsymbol{f} \in \boldsymbol{R}^{n_l+n_u}} \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2 + \infty \sum_{i}^{n_l} (f(\boldsymbol{x}_i) - y_i)^2$$

Naïve Solution

Right term solution: constrain f to match the supervised data

$$f(\boldsymbol{x}_i) = y_i \quad \forall\, i \in \{1, \ldots, n_l\}$$

Left term solution: enforce the solution f to be harmonic (cf. aggregation, rw)

$$f(\boldsymbol{x}_i) = \frac{\sum_{ij} f(\boldsymbol{x}_j) w_{ij}}{\sum_{ij} w_{ij}} \quad \forall\, i \in \{n_l + 1, \ldots, n_l + n_u\}$$

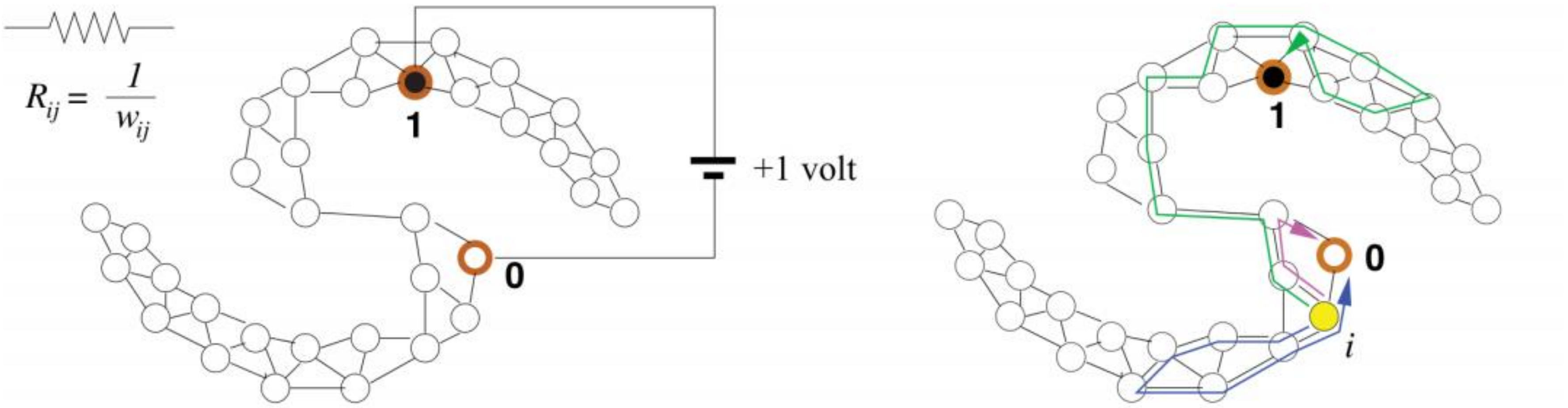
How can we handle unlabeled data?

SSL with Graphs: Harmonic Functions

Properties of the relaxation from ± 1 to $\mathbb{R}$

- There is a closed form solution for f
- this solution is unique
- globally optimal
- $f(\boldsymbol{x}_i)$ may not be integer
  - but we can threshold it
- electric-network interpretation
- random-walk interpretation

SSL with Graphs: Harmonic Functions



(a) The electric network interpretation (b) The random walk interpretation

Random walk interpretation :

1) start from the vertex you want to label and randomly walk
2) $P(j|i) = \frac{w_{ij}}{\sum_k w_{ik}} \Leftrightarrow \boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$
3) finish when a labeled vertex is hit

$$f(\boldsymbol{x}_i) = \frac{\sum_{ij} f(\boldsymbol{x}_j) w_{ij}}{\sum_{ij} w_{ij}}$$

4) $f(\boldsymbol{x}_i)$ is assigned by average of the labels of the hit vertices.

SSL with Graphs: Harmonic Functions

Iterative Solution: propagation

Step 1: Set f to match the supervised data

$$f(\boldsymbol{x}_i) = y_i \quad \forall\, i \in \{1, \ldots, n_l\}$$

Step 2: Propagate iteratively (only for unlabeled)

$$f(\boldsymbol{x}_i) \leftarrow \frac{\sum_{ij} f(\boldsymbol{x}_j) w_{ij}}{\sum_{ij} w_{ij}} \quad \forall\, i \in \{n_l + 1, \ldots, n_l + n_u\}$$

Properties:

- this will converge to the harmonic solution
- we can set the initial values for unlabeled nodes arbitrarily
- an interesting option for large-scale data

SSL with Graphs: Harmonic Functions

Closed form Solution:

Define $f_i \triangleq f(\boldsymbol{x}_i)$, $\boldsymbol{f} \triangleq [f_1, \dots, f_i, \dots, f_{n_l+n_u}]$

$$\Omega(\boldsymbol{f}) = \sum_{i,j=1}^{n_l+n_u} w_{ij}\left(f_i - f_j\right)^2 = \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$$

Then, $\boldsymbol{L}$ is a $(n_l+n_u) \times (n_l+n_u)$ matrix:

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{ll} & \boldsymbol{L}_{lu} \\ \boldsymbol{L}_{ul} & \boldsymbol{L}_{uu} \end{bmatrix}$$

The problem becomes

$$\min_{\boldsymbol{f}} \Omega(\boldsymbol{f}) (= \boldsymbol{f}_l^T \boldsymbol{L}_{ll} \boldsymbol{f}_l + \boldsymbol{f}_l^T \boldsymbol{L}_{lu} \boldsymbol{f}_u + \boldsymbol{f}_u^T \boldsymbol{L}_{ul} \boldsymbol{f}_l + \boldsymbol{f}_u^T \boldsymbol{L}_{uu} \boldsymbol{f}_u)$$

The solution can be obtained by

$$\nabla_{\boldsymbol{f}_u} \Omega(\boldsymbol{f}) = 2\boldsymbol{L}_{ul} \boldsymbol{f}_l + 2\boldsymbol{L}_{uu} \boldsymbol{f}_u = 0$$

$$\Rightarrow \boxed{\boldsymbol{f}_u = \boldsymbol{L}_{uu}^{-1}(-\boldsymbol{L}_{ul} \boldsymbol{f}_l) = \boldsymbol{L}_{uu}^{-1}(\boldsymbol{W}_{ul} \boldsymbol{f}_l)}$$

SSL with Graphs: Harmonic Functions

Relation between closed form solution and random work

$$\boldsymbol{f}_u = \boldsymbol{L}_{uu}^{-1}(\boldsymbol{W}_{ul}\boldsymbol{f}_l), \qquad \boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$$

Note that

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W} \rightarrow \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} = \boldsymbol{D}(\mathbf{I} - \boldsymbol{D}^{-1}\boldsymbol{W}) = \boldsymbol{D}(\mathbf{I} - \boldsymbol{P})$$

This yields

$$\boldsymbol{f}_u = (\mathbf{I} - \boldsymbol{P}_{uu})^{-1}\boldsymbol{D}_{uu}^{-1}(\boldsymbol{D}_{uu}\boldsymbol{P}_{ul}\boldsymbol{f}_l) = (\mathbf{I} - \boldsymbol{P}_{uu})^{-1}\boldsymbol{P}_{ul}\boldsymbol{f}_l$$

For $i \in \mathcal{U}$,

$$\begin{aligned} f_i &= (\mathbf{I} - \boldsymbol{P}_{uu})_{iu}^{-1}\boldsymbol{P}_{ul}\boldsymbol{f}_l \\ &= \textstyle\sum_{j:y_j=1}(\mathbf{I} - \boldsymbol{P}_{uu})_{iu}^{-1}\boldsymbol{P}_{uj} - \sum_{j:y_j=-1}(\mathbf{I} - \boldsymbol{P}_{uu})_{iu}^{-1}\boldsymbol{P}_{uj} \\ &= p_i^{(+1)} - p_i^{(-1)} \end{aligned}$$

SSL with Graphs: Regularized Harmonic Functions

For $i \in \mathcal{U}$,

$$f_i = p_i^{(+1)} - p_i^{(-1)} \Longrightarrow f_i = |f_i| sgn(f_i) = confidence \times label$$

What happens if an outlier sneaks in?

$\Longrightarrow$ The prediction for the outlier can mislead the confidence

How to control the confidence of the inference?

$\Longrightarrow$ Allow the random walk to die!

$\Longrightarrow$ We add a sink to the graph, where

sink = artificial node with label 0.

We connect the sink to every other vertex.

What will the sink do in the predictions?

SSL with Graphs: Regularized Harmonic Functions

Regularized Harmonic solution on the graph with sink

$$\nabla_{\boldsymbol{f}_u}\Omega(\boldsymbol{f}) = \mathbf{0}, \qquad \Omega(\boldsymbol{f}) = \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f}$$

$$\begin{bmatrix} \boldsymbol{L}_{ll} + \gamma_g \mathbf{I}_{n_l} & \boldsymbol{L}_{lu} & -\gamma_g \mathbf{1}_{n_l\times 1} \\ \boldsymbol{L}_{ul} & \boldsymbol{L}_{uu} + \gamma_g \mathbf{I}_{n_u} & -\gamma_g \mathbf{1}_{n_u\times 1} \\ -\gamma_g \mathbf{1}_{1\times n_l} & -\gamma_g \mathbf{1}_{1\times n_u} & (n_l + n_u)\gamma_g \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_l \\ \boldsymbol{f}_u \\ 0 \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{0}_u \\ \vdots \end{bmatrix}$$

We can disregard the last column and row:

$$\begin{bmatrix} \boldsymbol{L}_{ll} + \gamma_g \mathbf{I}_{n_l} & \boldsymbol{L}_{lu} \\ \boldsymbol{L}_{ul} & \boldsymbol{L}_{uu} + \gamma_g \mathbf{I}_{n_u} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_l \\ \boldsymbol{f}_u \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{0}_u \end{bmatrix}$$

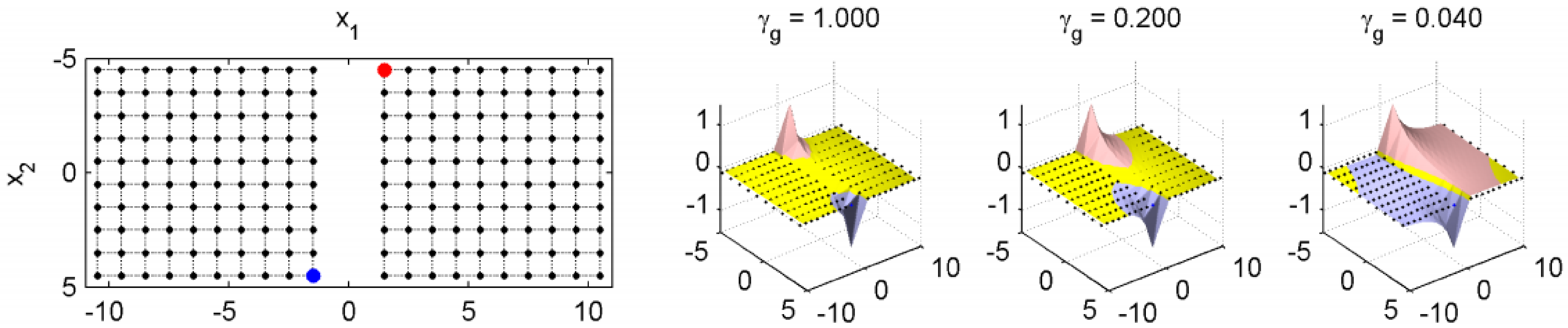
$$\Longrightarrow \boldsymbol{L}_{ul}\boldsymbol{f}_l + \left(\boldsymbol{L}_{uu} + \gamma_g \mathbf{I}_{n_u}\right)\boldsymbol{f}_u = \mathbf{0}_u$$

SSL with Graphs: Regularized Harmonic Functions

How do we compute this regularized random walk?

$$\boldsymbol{f}_u = (\boldsymbol{L}_{uu} + \gamma_g \mathbf{I})^{-1}(\boldsymbol{W}_{ul}\boldsymbol{f}_l),$$

How does γ_g influence the solution?



What happens to sneaky outliers?

SSL with Graphs: Soft Harmonic Functions

Regularized HS objective with $\boldsymbol{Q} = \boldsymbol{L} + \gamma_g \mathbf{I}$:

Define $f_i \triangleq f(\boldsymbol{x}_i)$, $\boldsymbol{f} \triangleq [f_i, \dots, f_{n_l+n_u}]$

$$\min_{\boldsymbol{f}\in\mathbb{R}^{n_l+n_u}} \infty \sum_{i=1}^{n_l} (f_i - y_i)^2 + \lambda \boldsymbol{f}^T \boldsymbol{Q} \boldsymbol{f}$$

Soft constraints for $f(\boldsymbol{x}_i) = y_i,\ \forall i \in \mathcal{L}$: ∞ is replaced by finite values

$$\min_{\boldsymbol{f}\in\mathbb{R}^{n_l+n_u}} (\boldsymbol{f} - \boldsymbol{y})^T \boldsymbol{C} (\boldsymbol{f} - \boldsymbol{y}) + \boldsymbol{f}^T \boldsymbol{Q} \boldsymbol{f}$$

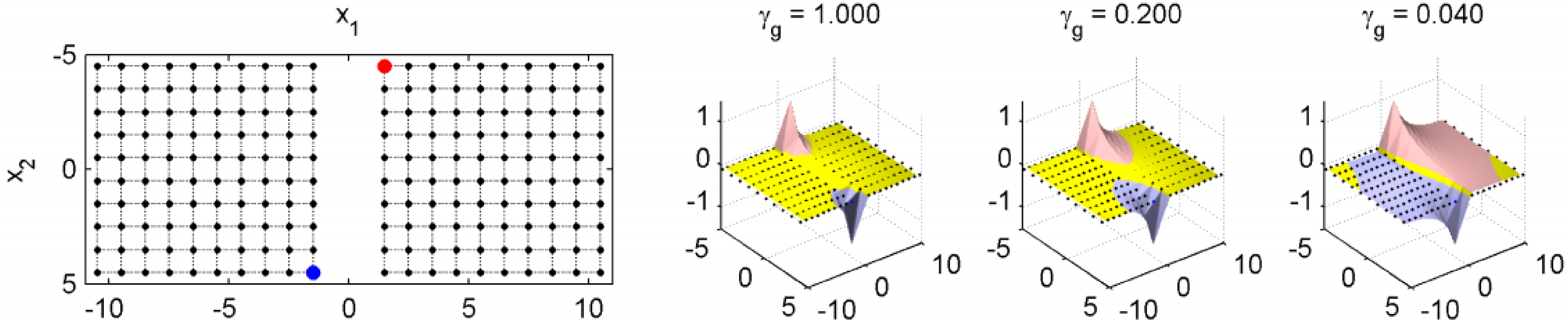
$\boldsymbol{C}$ is diagonal with $C_{ii} = \begin{cases} C_l & for\ \ labeled\ \ examples \\ C_u & for\ \ unlabeled\ \ examples \end{cases}$

$\boldsymbol{y}$ indicates pseudo-targets with $y_i = \begin{cases} true\ label & for\ \ labeled\ \ examples \\ 0 & for\ \ unlabeled\ \ examples \end{cases}$

SSL with Graphs: Soft Harmonic Functions

Closed form **Soft Harmonic solution:**

$$\boldsymbol{f}^* = \min_{\boldsymbol{f}\in\mathbb{R}^{n_l+n_u}} [(\boldsymbol{f}-\boldsymbol{y})^T \boldsymbol{C}(\boldsymbol{f}-\boldsymbol{y}) + \boldsymbol{f}^T \boldsymbol{Q}\boldsymbol{f}] \rightarrow \boldsymbol{f}^* = (\boldsymbol{C}^{-1}\boldsymbol{Q}+\mathbf{I})^{-1}\boldsymbol{y}$$



What are the differences between hard and soft?
Not much different in practice.
Noisy labels may be smoothed by soft harmonic.
Generalization is improved by the sink node

Summary questions on the lecture

- What does hinge loss in SVM penalize?
- What does hat loss in SVM for semi-supervised learning penalize?
- Why can’t we use eigenvectors to solve MinCut-based SSL in graph?
- What is the meaning of harmonic function in SSL?
- What is random work interpretation of harmonic function-based SSL in graph?
- What is a key point of regularized harmonic function-based SSL in graph?
- What is a key point of soft harmonic function-based SSL in graph?