Classification
Semi-supervised learning based on network

Speakers: Hanwen Wang, Xinxin Huang, and Zeyu Li

CS 249-2

2017 Winter

Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty

School of Computer Science, Carnegie Mellon University,
Gatsby Computational Neuroscience Unit, University College London

Introduction

Supervised Learning

labeled data is expensive

- Skilled human anotators
- Time consuming
- example: protein shape classficaton

Semi-supervised Learning

Exploit the manifold structure of data

Assumption: similar unlabeled data should be under one category

Frame Work

Annotation:

- Labeled points: L = {1,..,l}
- Unlabel points: U = {l+1,..,l+u}
- The similairty betwen point i and j: w(i,j)

Objective:

- Find a funtion: $f : V \to \mathbb{R}$
  such that the energy function $E(f) = \frac{1}{2}\sum_{i,j} w_{xy}(f(i) - f(j))^2$ is minimized.

- Similar points have higher weight

Derivation 1

How to find the minimum of a function?
Ans: first derivation

$$E(f) = \frac{1}{2}\sum_{i,j} w_{xy}(f(i) - f(j))^2$$

Partial derivatoin

$$\frac{\partial E(f)}{\partial f(i))} = \frac{1}{2}\sum_{j} w_{ij} * 2(f(i) - f(j))$$

$$= \sum_{j} (w_{ij} f(i) - w_{ij} f(j))$$

Assign the right hand size to zero gives us:

$$f(i) = \frac{\sum_j w_{ij} f(j)}{\sum_j w_{ij}}$$

$$= \frac{\sum_j w_{ij} f(i)}{d_i}$$

Derivation 2

Since $f = \arg\min_{f|_L = f_l} E(f)$ is harmonic, f satisfy $\Delta f = 0$

https://en.wikipedia.org/wiki/Harmonic_function

$$(D - W)f = 0$$

$$\begin{bmatrix} f_1 \\ f_2 \\ .. \\ f_n \end{bmatrix} = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & .. & \\ & & & d_4 \end{bmatrix}^{-1} \begin{bmatrix} w_{11} & .. & .. & w_{1n} \\ .. & & & .. \\ .. & & & .. \\ w_{n1} & .. & .. & w_{nn} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ .. \\ f_n \end{bmatrix}$$

If we pick a row and expand the matrix multiplication, we will get

$$f(i) = \frac{\sum_j w_{ij} f(i)}{d_i}$$

Derivation 3

Now, we do the calculation in matrix form

$$\Delta f = 0 \quad \longrightarrow \quad Df = Wf$$

$$\begin{pmatrix} D_{ll} & 0 \\ 0 & D_{uu} \end{pmatrix} \begin{pmatrix} f_l \\ f_u \end{pmatrix} = \begin{pmatrix} w_{uu} & w_{lu} \\ w_{ul} & w_{uu} \end{pmatrix} \begin{pmatrix} f_l \\ f_u \end{pmatrix}$$

Expanding the second row, we get:

$$D_{uu} f_u = w_{ul} f_l + w_{uu} f_u$$

$$f_u = (D_{uu} - w_{uu})^{-1} w_{ul} f_l$$

Since: $P = D^{-1} w \rightarrow w_{uu} = D_{uu} P_{uu}$

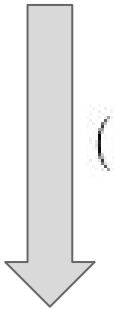
We get:

$$f_u = (I - P_{uu})^{-1} P_{ul} f_l$$

Derivation 4

Further expand the equation

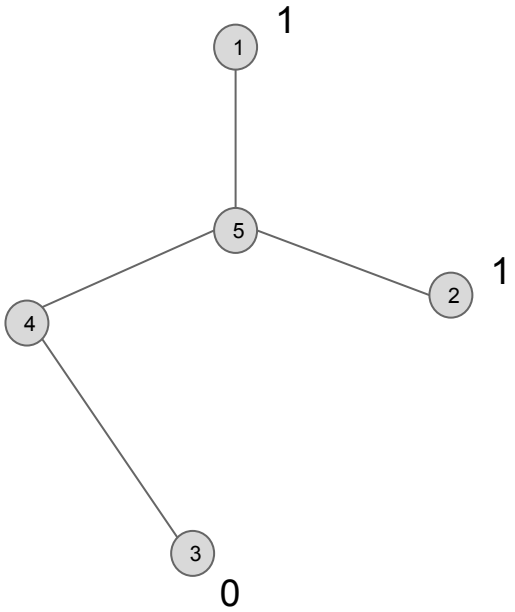
$$f_u = (I - P_{uu})^{-1} P_{ul} f_l$$



$$(I - A)^{-1} = (I + A + A^2 + A^3 + \ldots A^n)$$

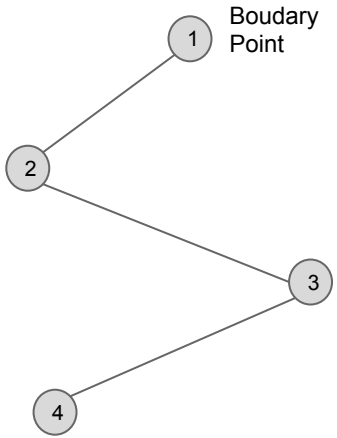
$$f_u = p_{ul} f_l + p_{uu} p_{ul} f_l + p_{uu}^2 p_{ul} f_l + \cdots p_{uu}^n p_{ul} f_l$$

Example



$$W = \begin{array}{c|ccccc} & x1 & x2 & x3 & x4 & x5 \\ x1 & 1.0 & 0.5 & 0.2 & 0.5 & 0.8 \\ x2 & 0.5 & 1.0 & 0.1 & 0.2 & 0.8 \\ x3 & 0.2 & 0.1 & 1.0 & 0.8 & 0.5 \\ \hline x4 & 0.5 & 0.2 & 0.8 & 1.0 & 0.8 \\ x5 & 0.8 & 0.8 & 0.5 & 0.8 & 1.0 \end{array}$$

Interpretation 1: Random Walk



$$P = \begin{array}{c|cccc} & x1 & x2 & x3 & x4 \\ \hline x1 & 0.0 & 0.0 & 0.0 & 0.0 \\ x2 & 0.5 & 0.0 & 0.5 & 0.0 \\ x3 & 0.0 & 0.5 & 0.0 & 0.5 \\ x4 & 0.0 & 0.0 & 1.0 & 0.0 \end{array}$$

Interpretation 2: Electric Network

Edges: resistor with conductance

Point Labels: voltage

P = V^2 / R

Energy dissipation is minimized since the voltage difference between two neighbors are minimized

Interpretation 3: Graph Kernels

- Heat equation: a parabolic partial differential equation that describes the distribution of heat (or variation in temperature) in a given region over time

$$\frac{\partial u_t(x,y)}{\partial t} = \alpha(\frac{(\partial^2 u_t(x,y)}{\partial^2 x} + \frac{(\partial^2 u_t(x,y)}{\partial^2 y}) = \alpha \Delta u_t(x,y)$$

- Heat kernel, it is a solution:

$$K_t = e^{-t\Delta}$$

$K_t(i,j)$: the solution of heat equation with initial conditions being a point source at i.

Interpretation 3: Graph Kernels

If we use this kernel in kernel classifier:

$$\hat{f}_t(j) = \sum_{i \in L} \alpha_i y_i K_t(i, j)$$

The kernel classifier can be considered as the solution of the heat equation with initial heat resource at the labeled data.

Interpretation 3: Graph Kernels

If we don’t consider the time and we only consider about the temprature relation between different points

$$G = \int_0^\infty K_t dt = \Delta^{-1}$$

Consider the green funtion on unlabel data

$$f_u = G_{uu} W_{ul} f_l \qquad\qquad f(i) = \sum_{i=1}^{l} \sum_k y_i w_{ik} G(k, j)$$

Our method can be interpreted as a kernel classifier with kernel G

Interpretation 3: Graph Kernels

- Spectrum of the G is the inverse of spectrum of Δ
  - This can indicate a connection to the work of Chapelle et al. 2002 about cluster kernels for semi-supervised learning.
  - By manipulating the eigenvalues of graph Laplacian, we can construct kernels which implement the cluster assumption: the induced distance depends on whether the points are in the same cluster or not.

Interpretation 4: Spectral Clustering

- Normalized cutting problem: Minimize the cost function:

$$min_f \frac{f^T(D-W)f}{f^T D f}$$

The solution is the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem

$$\Delta f = \lambda D f \quad \text{or} \quad (I - D^{-1}W)f = \lambda f$$

Spectral Clustering with group constraints

- Yu and Shi (2001) added group bias into the normalized cutting problem to specify which points should be in the same group.
- They proposed some pairwise grouping constraints of the labeled data.
- Imply the intuition that the points tend to be in the same cluster(have the labels) as its neighbors.

Label Propagation v.s.Constrainted Clustering

Semi-supervised learning on the graph can be interpreted in two ways

- In label propagation algorithms, the known labels are propagated to the unlabeled nodes.
- In constrained spectral clustering algorithms, known labels are first converted to pairwise constraints, then a constrained cut is computed as a tradeoff between minimizing the cut cost and maximizing the constraint satisfaction
- (Wang and Qian 2012)

Incorporating Class Prior Knowledge

- Decision rule is $f(i) > \frac{1}{2}$, assign to label 1, otherwise assign to label 0.
  - it works only when the classes are well separated. However, in real datasets ,the situation is different. Using f tend to produce severely unbiased classification
- Reason: W may be poorly estimated and does not reflect the classification goal.

We can not fully trust the the graph structure.

We want to incorprate the class prior knowledge in our model

Incorporating Class Prior Knowledge

- q: proportion for class 1; 1-q: proportion for class 0.
- To match this priors, we modified the decision rule by class mass normalization as

$$q\frac{f_u(i)}{\sum_i f_u(i))} > (1-q)\frac{1-f_u(i)}{\sum_i (1-f_u(i))}$$

Example: f = [0.1,0.2,0.3,0.4] and q = 0.5

L.H.: [0.05,0.1,0.15,0.2]

R.H.: [0.15,0.133,0.117,0.1]

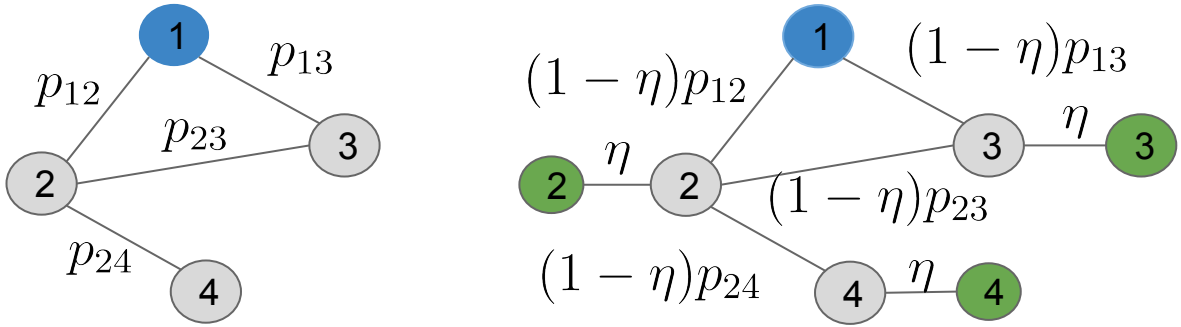
The first 2 will be assigned with label1,

while the last 2 will be assigned with label 0.

Incorporating Externel Classifier

- Assume the external classifer produces label h_u on the unlabeled data.
  - it can either be 0/1 or soft label [0.1]

$$f_u = (I - (1-\eta)P_{uu})^{-1}((1-\eta)P_{ul}f_l + \eta h_u)$$



Learning the Weight Matrix W

Recall the defination of Weight Matrix

$$w_{ij} = exp(-\sum_{d=1}^{m} \frac{(x_{jd} - x_{id})^2}{\sigma_d^2})$$

This will be a feature selection mechanism which better aligns the graph structure with the data.

Learn σ_d by minimizing the average label entropy

$$H(f) = \frac{1}{u} \sum_{i=l+1}^{l+u} H_i(f(i))$$

Learning the Weight Matrix W

$$H_i(f(i)) = -f(i)log(f(i)) - (1 - f(i))log(1 - f(i))$$

Why we can get optimal σ_d by minimizing H?

- Small H(i) implies that f(i) is close to 0 or 1.
- This captures intuition that a good W (equivalently a good set of { σ_d}) should result in confident labeling.
- min H lead to a set of optimal σ_d which can result in confident labeling u.

Learning the Weight Matrix W

- Important property of H is that H has a minimum at 0 as $\sigma_d \to 0$
- Solution:

$$\widetilde{P} = \epsilon U + (1 - \epsilon) P$$

The label will not be dominated by its nearest neighbor. It can also be influenced by all the other nodes.

- Use the gradient decsent to get the hyperparameter σ_d

Conclusion

- Harmonic function is strong model to solve the semi-supervised learing problem.
- Label propagation and constrained spectral clustering algorithms can also be implemented to solve the semi-supervised learning tasks.
- This model is flexible and can be easily incorprated with external helpful information.

Graph Regularized Transductive Classification on Heterogeneous Information Networks

Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han and Jing Gao
Dept. of Computer Science,
University of Illinois at Urbana-Champaign

Introduction

Semi-supervised Learning:

Classify the unlabeled data based on known information

Two groups of classification:

Transductive classification - to predict labels for the given unlabeled data

Inductive classification - construct decision function in whole data space

Homogeneous network & **Heterogeneous network**

Classifying multi-typed objects into classes.

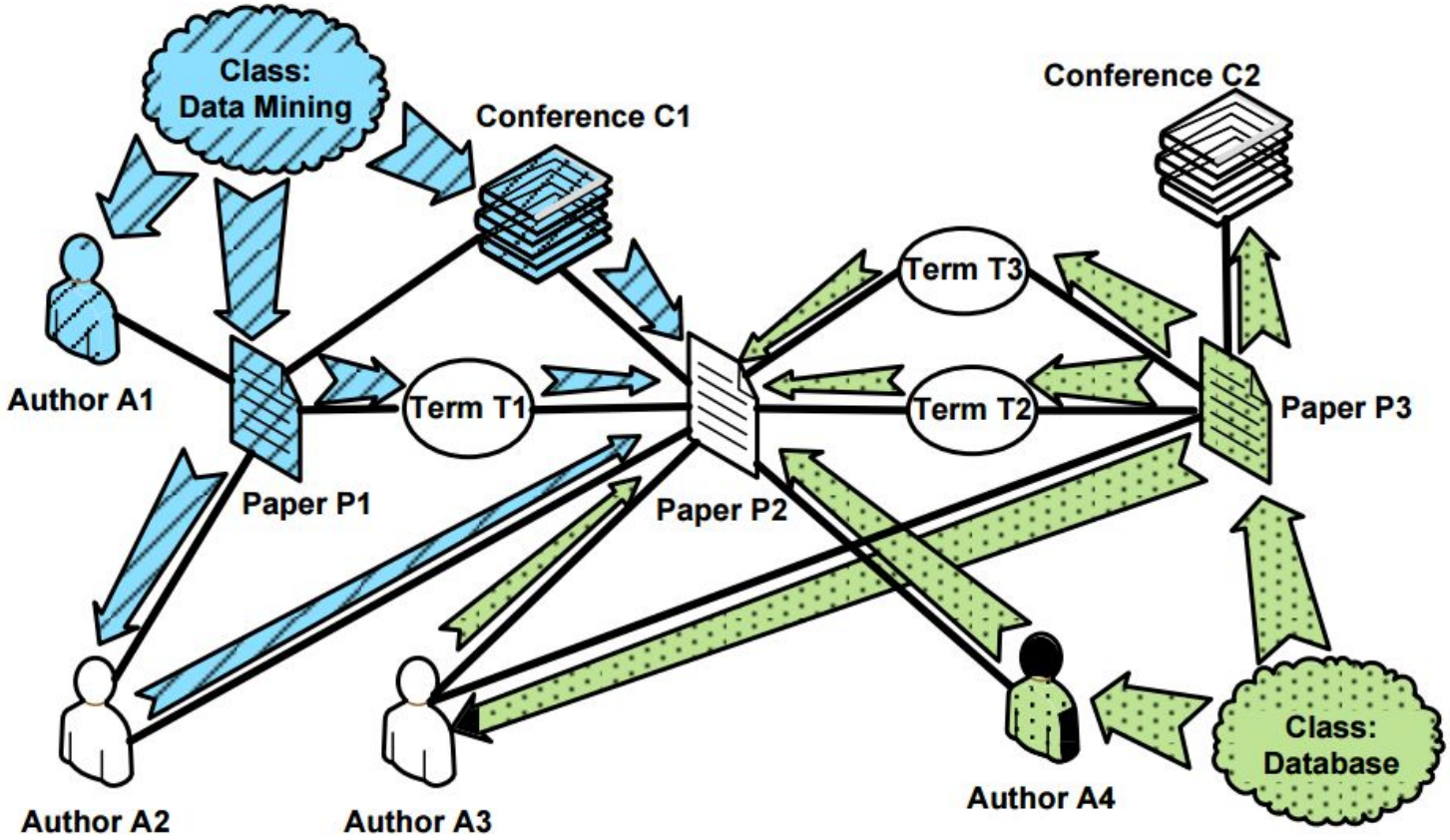


Fig. 1. Knowledge propagation in a bibliographic information network

Problem Definition

Definition 1: Heterogeneous information network

m types of data objects: $\mathcal{X}_1 = \{x_{11}, \ldots, x_{1n_1}\}, \ldots, \mathcal{X}_m = \{x_{m1}, \ldots, x_{mn_m}\}$

Graph $G =< V, E, W >$

$V = \cup_{i=1}^{m} \mathcal{X}_i \quad m \geq 2$

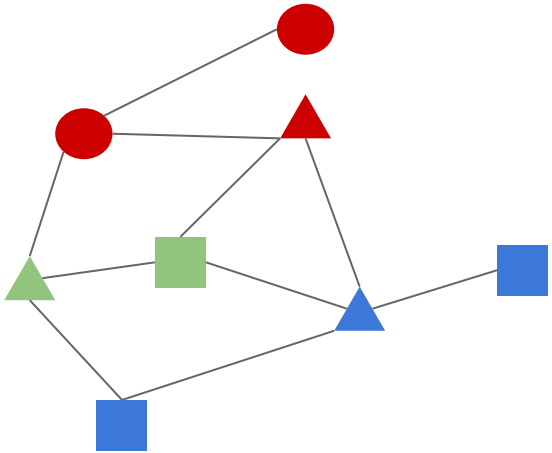
Problem Definition

Definition 2: Class

Given $G =< V, E, W >$ and $V = \cup_{i=1}^{m} \mathcal{X}_i$,

Class: $G' =< V', E', W' >$, where $V' \subseteq V$, $E' \subseteq E$.

$W'_{x_{ip}x_{jq}} = W_{x_{ip}x_{jq}} \ \forall e =< x_{ip}, x_{jq} > \in E'$



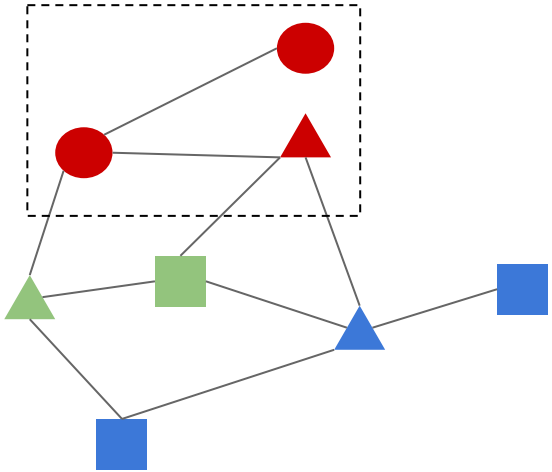
Problem Definition

Definition 2: Class

Given $G =< V, E, W >$ and $V = \cup_{i=1}^{m} \mathcal{X}_i$,

Class: $G' =< V', E', W' >$, where $V' \subseteq V$, $E' \subseteq E$.

$W'_{x_{ip}x_{jq}} = W_{x_{ip}x_{jq}} \ \forall e =< x_{ip}, x_{jq} > \in E'$

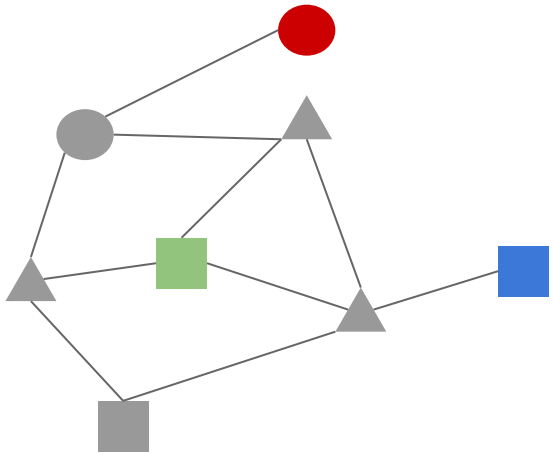


Problem Definition

Definition 3: Transductive classification on heterogeneous information networks

Given $G =< V, E, W >$ and $V' \subseteq V$ which are labeled with value $\mathcal{Y}$,

Predict the class labels for all unlabeld object $V - V'$

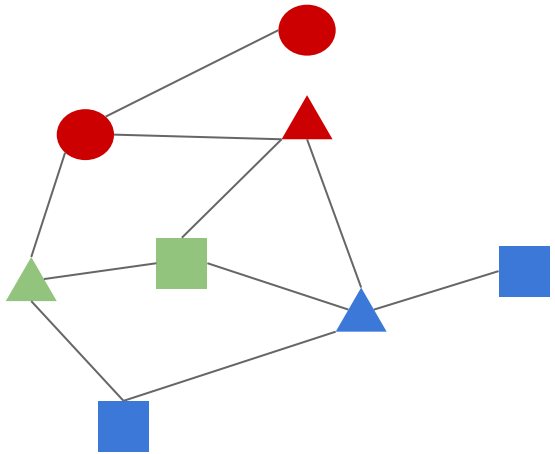


Problem Definition

Definition 3: Transductive classification on heterogeneous information networks

Given $G =< V, E, W >$ and $V' \subseteq V$ which are labeled with value $\mathcal{Y}$,

Predict the class labels for all unlabeld object $V - V'$



Problem Definition

Suppose the number of classifiers is K

Compute $\boldsymbol{F}_i = [\boldsymbol{f}_i^{(1)}, \ldots, \boldsymbol{f}_i^{(K)}] \in \mathbb{R}^{n_i \times K}$, where each $\boldsymbol{f}_i^k = [f_{i1}^k, \ldots, f_{in_i}^k]$ measures the confidence of $x_{ip} \in \mathcal{X}_i$ belongs to class k.

Class of x_{ip} is $\arg \max_{1 \le k \le K} f_{ip}^{(k)}$.

Use $\boldsymbol{R}_{ij}$ to denote the relation matrix of Type i and Type j, $\boldsymbol{R}_{ij} \in \mathbb{R}^{n_i \times n_j}$.

$R_{ij,pq}$ represents the weight on link $< x_{ip}, x_{jq} >$

$$R_{ij,pq} = \begin{cases} 1 & \text{if data objects } x_{ip} \text{ and } x_{jq} \text{ are linked together} \\ 0 & \text{otherwise} \end{cases}$$

Problem Definition

Another vector to use:

$$y_{ip} = \begin{cases} 1 & \text{if } x_{ip} \text{ is labeled to the k-th class} \\ 0 & \text{otherwise} \end{cases}$$

The goal is to predict infer a set of $\boldsymbol{f}_i^k$ from $\boldsymbol{R}_{ij}$ and $y_i^{(k)}$.

Graph-based Regularization Framework

Intuition:

Prior Knowledge: A1, P1 and C1 belong to “data mining” => Infer: A2, T1 are highly related to data mining.

Similarly: A3, C2, T2, and T3 highly related to “database”.

Knowledge propagation.

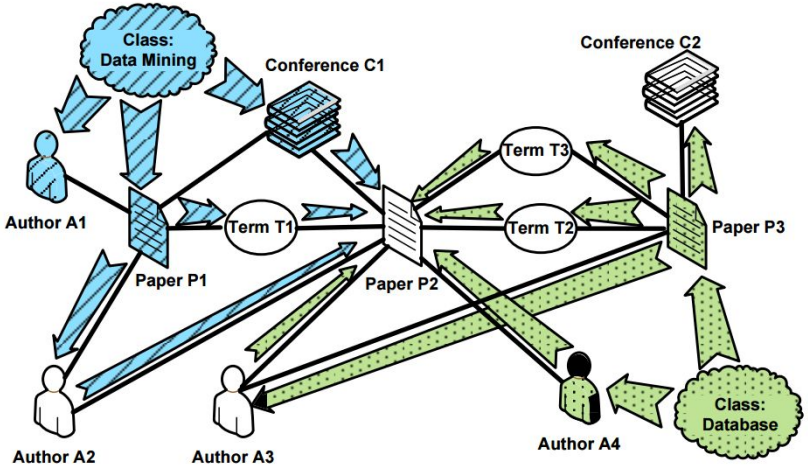


Fig. 1. Knowledge propagation in a bibliographic information network

Graph-based Regularization Framework

Formulate Intuition as follows:

(1) The estimated confidence measure of two objects x_{ip} and x_{jq} belonging to class k, $f_{ip}^{(k)}$ and $f_{jq}^{(k)}$, should be similar if x_{ip} and x_{jq} are linked together, i.e., the weight value $R_{ij,pq} > 0$.

(2) The confidence estimation $\boldsymbol{f}_i^k$ should be similar to the ground truth, $\boldsymbol{y}_i^{(k)}$.

Graph-based Regularization Framework

The Algorithm:

Define a diagonal matrix $\boldsymbol{D}_{ij}$ of size $n_i \times n_i$. The (p,p)-th of $\boldsymbol{D}_{ij}$ is the sum of the p-th row of $\boldsymbol{R}_{ij}$.

Objective function:

$$J(\boldsymbol{f}_i^{(k)}, \ldots, \boldsymbol{f}_m^{(k)}) = \sum_{i,j=1}^{m} \lambda_{i,j} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} R_{ij,pq} \left(\frac{1}{\sqrt{D_{ij,pp}}} f_{ip}^{(k)} - \frac{1}{\sqrt{D_{ji,qq}}} f_{jq}^{(k)} \right)^2 + \sum_{i=1}^{m} \alpha_i (\boldsymbol{f}_i^{(k)} - \boldsymbol{y}_i^{(k)})^T (\boldsymbol{f}_i^{(k)} - \boldsymbol{y}_i^{(k)})$$

Graph-based Regularization Framework

Trade-off:

Controlled by λ_{ij} and α_i where

Larger λ_{ij}: more rely on relationship of $\mathcal{X}_i$ and $\mathcal{X}_j$.

Larger α_i: The label of i is more trustworthy.

Prior Knowledge.

Define normalized form $\boldsymbol{S}_{ij} = \boldsymbol{D}_{ij}^{(-1/2)} \boldsymbol{R}_{ij} \boldsymbol{D}_{ji}^{-1/2}$

Graph-based Regularization Framework

Rewrite the objective function as

$$J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)}) = \sum_{i,j=1}^{m} \lambda_{ij}((\boldsymbol{f}_i^{(k)})^T \boldsymbol{f}_i^{(k)} + (\boldsymbol{f}_j^{(k)})^T \boldsymbol{f}_j^{(k)} - 2(\boldsymbol{f}_i^{(k)})^T \boldsymbol{S}_{ij} \boldsymbol{f}_j^{(k)}) + \sum_{i=1}^{m} \alpha_i (\boldsymbol{f}_i^{(k)} - \boldsymbol{y}_i^{(k)})^T (\boldsymbol{f}_i^{(k)} - \boldsymbol{y}_i^{(k)})$$

Reduce to

$$J(\boldsymbol{f}_1^{(k)}) = 2\lambda_{11}(\boldsymbol{f}_1^{(k)})^T \mathbf{L}_{11} \boldsymbol{f}_1^{(k)} + \alpha_1 (\boldsymbol{f}_1^{(k)} - \mathbf{y}_1^{(k)})^T (\boldsymbol{f}_1^{(k)} - \mathbf{y}_1^{(k)})$$

in homogeneous information networks. $\mathbf{L}_{ii} = \mathbf{I}_i - \mathbf{S}_{ii}$ is the *normalized graph Laplacian*.

Graph-based Regularization Framework

Given the following definition:

$$\boldsymbol{f}^{(k)} = [(\boldsymbol{f}_1^{(k)})^T, \ldots, (\boldsymbol{f}_m^{(k)})^T]^T \; ; \mathbf{y}^{(k)} = [(\mathbf{y}_1^{(k)})^T, \ldots, (\mathbf{y}_m^{(k)})^T]^T$$

$$\boldsymbol{\alpha} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_m) \qquad \mathbf{L}_{ij} = \begin{bmatrix} \mathbf{I}_i & -\mathbf{S}_{ij} \\ -\mathbf{S}_{ji} & \mathbf{I}_j \end{bmatrix}, \text{ where } i \neq j$$

Let $\boldsymbol{H}_{ij}$ be the $n \times n$ matrix, $\sum_{i=1}^{m} n_i = n$ And $\mathbf{H} = \sum_{i \neq j} \lambda_{ij} \mathbf{H}_{ij} + 2 \sum_{i=1}^{m} \lambda_{ii} \mathbf{H}_{ii}$

We can rewrite the objective function as:

$$J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)}) = (\boldsymbol{f}^{(k)})^T \mathbf{H} \boldsymbol{f}^{(k)} + \left(\boldsymbol{f}^{(k)} - \mathbf{y}^{(k)}\right)^T \boldsymbol{\alpha} \left(\boldsymbol{f}^{(k)} - \mathbf{y}^{(k)}\right)$$

Graph-based Regularization Framework

Solution

- **Closed form solution**

Hessian matrix of $H\left(J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)})\right) = 2\mathbf{H} + 2\boldsymbol{\alpha}$ is positive semi-definite.

$$\frac{\partial J}{\partial (\boldsymbol{f}_i^{(k)})^T} = \sum_{j=1, j\neq i}^{m} \lambda_{ij}(2\boldsymbol{f}_i^{(k)} - 2\mathbf{S}_{ij}\boldsymbol{f}_j^{(k)}) + 4\lambda_{ii}\mathbf{L}_{ii}\boldsymbol{f}_i^{(k)} + 2\alpha_i(\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)})$$

And setting $\frac{\partial J}{\partial (\boldsymbol{f}_i^{(k)})^T} = 0$ for all i.

$$\boldsymbol{f}_i^{(k)} = \Big((\sum_{j=1, j\neq i}^{m} \lambda_{ij} + \alpha_i)\mathbf{I}_i + 2\lambda_{ii}\mathbf{L}_{ii} \Big)^{-1} \Big(\alpha_i \mathbf{y}_i^{(k)} + \sum_{j=1, j\neq i}^{m} \lambda_{ij}\mathbf{S}_{ij}\boldsymbol{f}_j^{(k)} \Big) \ , \ i \in \{1, \ldots, m\}$$

- **Iterative solution**

Graph-based Regularization Framework

Solution

- **Closed form solution**
- **Iterative solution**

Step 0: Initialization. $\boldsymbol{f}_i^{(k)}(0) = \mathbf{y}_i^{(k)}$

Step 1: Based on current $\boldsymbol{f}_i^{(k)}(t)$, compute the

$$\boldsymbol{f}_i^{(k)}(t+1) = \frac{\sum_{j=1,j\neq i}^{m} \lambda_{ij}\mathbf{S}_{ij}\boldsymbol{f}_j^{(k)}(t) + 2\lambda_{ii}\mathbf{S}_{ii}\boldsymbol{f}_i^{(k)}(t) + \alpha_i \mathbf{y}_i^{(k)}}{\sum_{j=1,j\neq i}^{m} \lambda_{ij} + 2\lambda_{ii} + \alpha_i}$$

Step 2: Repeat Step 1 until converge. ($\boldsymbol{f}_i^{(k)}(t)$ change little over t-th iteration)

Step 3: Assign class label to p-th object of $\mathcal{X}_i$ by $c_{ip} = \arg\max_{1\leq k\leq K} f_{ip}^{(k)*}$.

Graph-based Regularization Framework

Complexity

- **Iterative solution** $O(NK(|E| + |V|))$

N: # of iterations. K: # of classes.

- **Closed form solution**

Worse than the iterative solution since iterative solution bypass the matrix inversion operation.

Q&A.

Hanwen Wang: hanwenwang@g.ucla.edu
Xinxin Huang: xinxinh@ucla.edu
Zeyu Li: zyli@cs.ucla.edu