Gossip-based Truth Discovery

Zhiying Xu

June 4, 2017

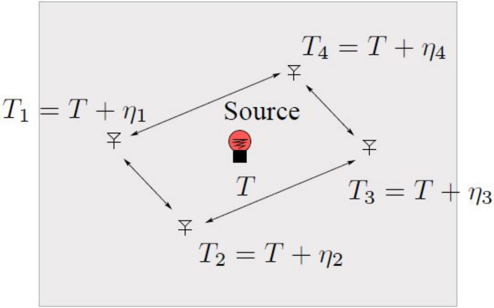
Outline

Outline

Outline

Gossip Algorithm

Gossip algorithms: schemes which distribute the computation burden and in which a node communicates with a randomly chosen neighbor.



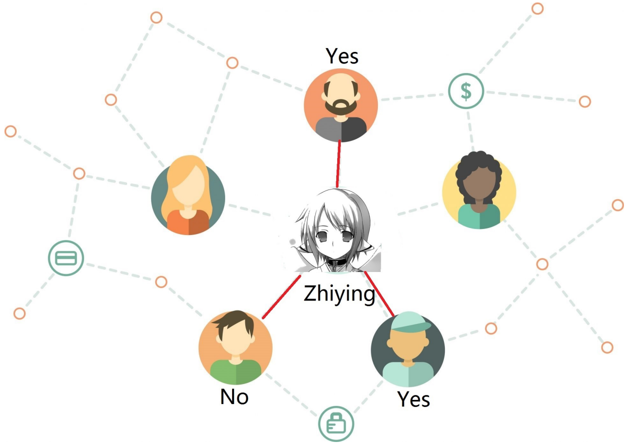
Truth Discovery

- Discover truth from noisy crowd sensing data
- Evaluate the reliability of sensors

Outline

A Real-life Case

Whether the class is canceled?



Outline

Model

Definition (Problem Formulation)

Given a crowd sensing model where N sensors make individual observations $\boldsymbol{X}^{N\times M} \in \{0,1\}^{N\times M}$ about a set of M variables, determine the true value of these variables $\boldsymbol{t}^M \in \{0,1\}^M$ which satisfies

$$\langle \boldsymbol{t}, \boldsymbol{r} \rangle = \underset{\langle \boldsymbol{t}, \boldsymbol{r} \rangle}{\arg\min}\, p\left(\boldsymbol{X} | \boldsymbol{t}, \boldsymbol{r}\right),$$

where $\boldsymbol{r}^N \in [0,1]$ is the reliability of sensors.

Model

Definition (Problem Formulation)

Given a crowd sensing model where N sensors make individual observations $\boldsymbol{X}^{N\times M} \in \{0,1\}^{N\times M}$ about a set of M variables, determine the true value of these variables $\boldsymbol{t}^M \in \{0,1\}^M$ which satisfies

$$\langle \boldsymbol{t}, \boldsymbol{r} \rangle = \arg\min_{\langle \boldsymbol{t}, \boldsymbol{r} \rangle} p\left(\boldsymbol{X} | \boldsymbol{t}, \boldsymbol{r}\right),$$

where $\boldsymbol{r}^N \in [0,1]$ is the reliability of sensors.

- $\boldsymbol{t}, \boldsymbol{r}$ are related to each other.
- There is no need for iteration by EM algorithm.

Objective Function

Using the relationship between $\boldsymbol{t}$ and $\boldsymbol{r}$, We can simplify the previous problem into the optimization of the objective function

$$\boldsymbol{t} = \arg\min_{\boldsymbol{t}} \sum_{i=1}^{N} f\left(\boldsymbol{x}_i, \boldsymbol{t}\right),$$

where $f\left(\boldsymbol{x}_i, \boldsymbol{t}\right)$ denotes
$\frac{||\boldsymbol{x}_i - \boldsymbol{t}||}{M} \ln \frac{M}{||\boldsymbol{x}_i - \boldsymbol{t}||} + \left(1 - \frac{||\boldsymbol{x}_i - \boldsymbol{t}||}{M}\right) \ln \left(1 - \frac{M}{||\boldsymbol{x}_i - \boldsymbol{t}||}\right).$

Truth Discovery Strategy

- All sensors are in a connected networks.
- Sensors have limited memory and can't store much data.
- Each sensor communicates with a random neighbor asynchronously.

Outline

NP Hardness of the Decision Version

Given a matrix $\boldsymbol{X}^{N\times M}$, is there a vector $\boldsymbol{t}^{M}$ which satisfies $\sum_{i=1}^{N} f\left(\boldsymbol{x}_i, \boldsymbol{t}\right) < C$?

Proof of NP Hardness

Theorem

Discovering the truth $\boldsymbol{t}$ *from the observed matrix* $\boldsymbol{X}$ *is NP-hard.*

- By reduction from 3 exact cover problem.
- Inspired by Cardinal, J., Fiorini, S. and Joret, G., 2012. Minimum entropy combinatorial optimization problems. Theory of Computing Systems, 51(1), pp.4-21.

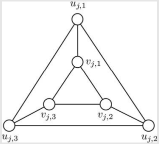
Proof of NP Hardness

Proof: Step 1

3 exact cover: Given a set system (U, S), decide whether U can be covered using pairwise disjoint sets from S

$$\Downarrow$$

minimum orientation defined on a concave function: Construct a graph by creating a gadget for each element in U.



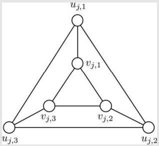
Given the graph, decide whether there is an orientation that satisfies $\sum_{i=1}^{|V|} g\left(\frac{d_i^+}{|E|}\right) < C_1$.

Here $g(\cdot)$ is a concave function and d_i^+ is the in-degree of the node i.

Proof of NP Hardness

Proof: Step 2

minimum orientation defined on a concave function:



Given the graph, decide whether there is an orientation that satisfies $\sum_{i=1}^{|V|} g\left(\frac{d_i^+}{|E|}\right) < C_1$. Here $g(\cdot)$ is a concave function and d_i^+ is the in-degree of the node i.

$\Downarrow$

Truth Discovery:
Covert the graph into a matrix $\boldsymbol{X}$, decide whether there is a vector $\boldsymbol{t}^M$ which satisfies $\sum_{i=1}^{N} f(\boldsymbol{x}_i, \boldsymbol{t}) < C_2$.
$\boldsymbol{X}$ = ancillary part + graph part.

Outline

Outline

The Complexity of Exact Algorithm

Theorem

The complexity of the exact algorithm is $\Theta\left(2^{rank(\boldsymbol{X})}\right)$.

The Complexity of Exact Algorithm

Theorem

The complexity of the exact algorithm is $\Theta\left(2^{rank(\boldsymbol{X})}\right)$.

- The base vectors can be obtained in polynomial time.
- The truth vector $\boldsymbol{t}$ is a combination of the base vectors of the observed vectors $\boldsymbol{x}_i$.

Exact Algorithm

Algorithm 1 Exact Algorithm

Input: Observed matrix $\boldsymbol{X}$

Output: Truth vector $\boldsymbol{t}$

Initialize: $S_1 = 0, S_2 = N$

Calculate the base vectors of $\boldsymbol{X}$

for $\boldsymbol{t}^* \in$ combination of the base vectors **do**

  for $i = 1$ to n **do**

    $S_1 = S_1 + f(\boldsymbol{x}_i, \boldsymbol{t}^*)$

  end for

  if $S_1 < S_2$ **then**

    $S_2 = S_1, \boldsymbol{t} = \boldsymbol{t}^*$

  end if

  $S_1 = 0$

end for

Outline

Approximation Algorithm

Theorem

The approximation algorithm achieves an approximation ratio of 1.7.

Approximation Algorithm

Theorem

The approximation algorithm achieves an approximation ratio of 1.7.

- The approximation ratio is an upper bound, for the worse case is difficult to find.
- Use triangle inequality $f(\boldsymbol{x}, \boldsymbol{y}) + f(\boldsymbol{y}, \boldsymbol{z}) > f(\boldsymbol{x}, \boldsymbol{z})$.
- The complexity is $\Theta\left(n^2\right)$

Approximation Algorithm

Algorithm 2 Approximation Algorithm

Input: Observed matrix $\boldsymbol{X}$
Output: Truth vector $\boldsymbol{t}$

Initialize: $S = 0, \boldsymbol{t}_2 = 0$
Find the 1-median point $\boldsymbol{t}_1$ in metric space $\left(\{0,1\}^M, f\right)$
for $i = 1$ to n **do**
  if $||\boldsymbol{x}_i - \boldsymbol{t}_1||_1/M < \frac{1}{4}$ **then**
    $\boldsymbol{t}_2 = \boldsymbol{t}_2 + \boldsymbol{x}_i$
  else if $||\boldsymbol{x}_i - \boldsymbol{t}_1||_1/M > \frac{3}{4}$ **then**
    $\boldsymbol{t}_2 = \boldsymbol{t}_2 + \boldsymbol{1} - \boldsymbol{x}_i$
  else
    $\boldsymbol{t}_2 = \boldsymbol{t}_2 + \frac{1}{2}\boldsymbol{1}$
  end if
end for
$\boldsymbol{t}_2 = round\,(\boldsymbol{t}_2/N)$
if $\sum_{i=1}^{N} f(\boldsymbol{x}_i, \boldsymbol{t}_1) > \sum_{i=1}^{N} f(\boldsymbol{x}_i, \boldsymbol{t}_2)$ **then**
  $\boldsymbol{t} = \boldsymbol{t}_2$
else
  $\boldsymbol{t} = \boldsymbol{t}_1$
end if

Outline

Outline

Decentralized Exact Algorithm

- It's hard to calculate the base vectors in a decentralized ways, so we decide to try every vector in $\{0,1\}^M$
- There is a trade-off between run time and accuracy.

$$\text{run time: } 2^M \frac{3\log \epsilon^{-1}}{\log \lambda_2 (W_1)^{-1}}$$

$$W_1 = I - \frac{1}{2n} D + \frac{P + P^T}{2n}$$

$$D_i = \sum_{j=1}^{n} [P_{ij} + P_{ji}]$$

$$\text{accuracy: } 1 + 2\epsilon$$

Decentralized Exact Algorithm

Algorithm 3 Decentralized Exact Algorithm

Input: Observed matrix $\boldsymbol{X}$
Output: Truth vector $\boldsymbol{t}$

```
Initialize: f_{min1} = N
for t' ∈ {0,1}^M do
    Every node calculates f_i = f(x_i, t')
    for k = 1 to (3 log ε^{-1}) / (log λ_2(W)^{-1}) do
        A node calculates the average f with it random neighbor
        and renew the values of both nodes
    end for
    if f_i < f_min then
        f_min = f_i, t = t'
    end if
end for
```

A Fast Algorithm for 1-median Problem

- Using triangle inequality in metric space and Chebyshev Inequility
- Inspired by Indyk, P., 1999, May. Sublinear time algorithms for metric space problems. In Proceedings of the thirty-first annual ACM symposium on Theory of computing (pp. 428-434). ACM.

A Fast Algorithm for 1-median Problem

Algorithm 4 Fast Algorithm for 1-median Problem

Input: Observed matrix $\boldsymbol{X}$

Output: 1-median Point $\boldsymbol{x}^*$

Initialize: $\boldsymbol{Y} = \boldsymbol{X}, \boldsymbol{Z} = [0]$

for $k = 1$ to $\log^2 n 4 \left(4/\sigma + 1\right)^2 / \sigma^2$ **do**

  Every node changes $\boldsymbol{y}_i$ with a random neighbor asynchronously.

  For every mixing time $\frac{\log n + \log \epsilon^{-1}}{1 - \lambda_P}$,
  $Z_{i||x_i - y_i||_1} = Z_{i||x_i - y_i||_1} + 1$

end for

for $k = 1$ to $8 \left(\ln 2 + 2 \ln n + \ln \epsilon^{-1}\right) / \Phi$ **do**

  For a node i compare with its random neighbor j,
  We only compare the sum of distance shorter than $4 f\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) / \sigma$

  $\boldsymbol{x}_i = \boldsymbol{x}_j =$ the smaller one

end for

$\boldsymbol{x}^* = \boldsymbol{x}_i$

Decentralized Approximation Algorithm

- Implement the fast algorithm for 1-median problem.
- There is a trade-off between run time and accuracy.

$$\text{run time: } \frac{6\log\epsilon_1^{-1}}{\log\lambda_2(W_1)^{-1}} + \frac{\log^2 n4(4/\sigma+1)^2}{\sigma^2}\frac{\log n+\log\epsilon_2^{-1}}{1-\lambda_2(W_2)}$$

$$8\frac{\ln 2+2\ln n+\ln\epsilon^{-1}}{\Phi}$$

$$W_2 = I - \frac{1}{n}D + \frac{P+P^T}{n}$$

$$D_i = \sum_{j=1}^{n}\left[P_{ij}+P_{ji}\right]$$

$$\text{accuracy: } \max\left\{1.5+4\epsilon_1, (1+\epsilon_2+\sigma)\frac{f(1/15/(1+\epsilon_2+\sigma))}{f(1/30/(1+\epsilon_2+\sigma))}\right\}$$

Decentralized Approximation Algorithm

Algorithm 5 Decentralized Exact Algorithm

Input: Observed matrix $\boldsymbol{X}$

Output: Truth vector $\boldsymbol{t}$

Initialize: $\boldsymbol{Y} = \boldsymbol{X}, \boldsymbol{Z} = [0]$
Implement the fast algorithm for 1-median problem and get the $\boldsymbol{x}^*$.
For every node i
if $||\boldsymbol{y}_i - \boldsymbol{x}^*||_1/M < \frac{1}{4}$ **then**
  $\boldsymbol{z}_i = \boldsymbol{y}_i$
else if $||\boldsymbol{x}_i - \boldsymbol{t}_1||_1/M > \frac{3}{4}$ **then**
  $\boldsymbol{z}_i = \mathbf{1} - \boldsymbol{y}_i$
else
  $\boldsymbol{z}_i = \frac{1}{2}\mathbf{1}$
end if
Calculate the average of $\boldsymbol{z} = round\left(\boldsymbol{z}_i\right)$
$\boldsymbol{t} = \arg\min_{\boldsymbol{x}^*, \boldsymbol{z}} \sum_{i=1}^{N} f\left(\boldsymbol{x}_i, \boldsymbol{t}\right)$

Outline

Convex Optimization of the Matrix

$$
\begin{aligned}
\text{minimize } & \lambda_2(W) \\
\text{subject to } & W = \sum_{i,j=1}^{n} \frac{1}{n} P_{ij} W_{ij} \\
& P \leq [0], \\
& P_{ij} = 0 \text{ if } \{i,j\} \notin E, \\
& \sum_j P_{ij} = 1, \forall i.
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
\text{minimize } & s \\
\text{subject to } & W - \mathbf{1}\mathbf{1}^T/n \preceq sI \\
& W = \sum_{i,j=1}^{n} \frac{1}{n} P_{ij} W_{ij} \\
& P \leq [0], \\
& P_{ij} = 0 \text{ if } \{i,j\} \notin E, \\
& \sum_j P_{ij} = 1, \forall i.
\end{aligned}
$$

Semidefinite Program(SDP)

Outline

Future Work

- Simulations on real data(cooperate with Jiefeng Li)
- Submit to INFOCOM 2018

Boyd, S., Ghosh, A., Prabhakar, B. and Shah, D., 2005, March. Gossip algorithms: Design, analysis and applications. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (Vol. 3, pp. 1653-1664). IEEE.

Wang, D., Kaplan, L., Le, H. and Abdelzaher, T., 2012, April. On truth discovery in social sensing: A maximum likelihood estimation approach. In Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on (pp. 233-244). IEEE.

Indyk, P., 1999, May. Sublinear time algorithms for metric space problems. In Proceedings of the thirty-first annual ACM symposium on Theory of computing (pp. 428-434). ACM.

Cardinal, J., Fiorini, S. and Joret, G., 2012. Minimum entropy combinatorial optimization problems. Theory of Computing Systems, 51(1), pp.4-21.

Thank You!