Active Learning for Streaming Networked Data

Zhilin Yang, Jie Tang, Yutao Zhang

Computer Science Department, Tsinghua University

Introduction

Mining streaming data becomes an important topic.

- Challenge 1: the lack of labeled data

  Related work: ***active learning for streaming data*** [28, 6, 5, 29]

- Challenge 2: network correlation between data instances

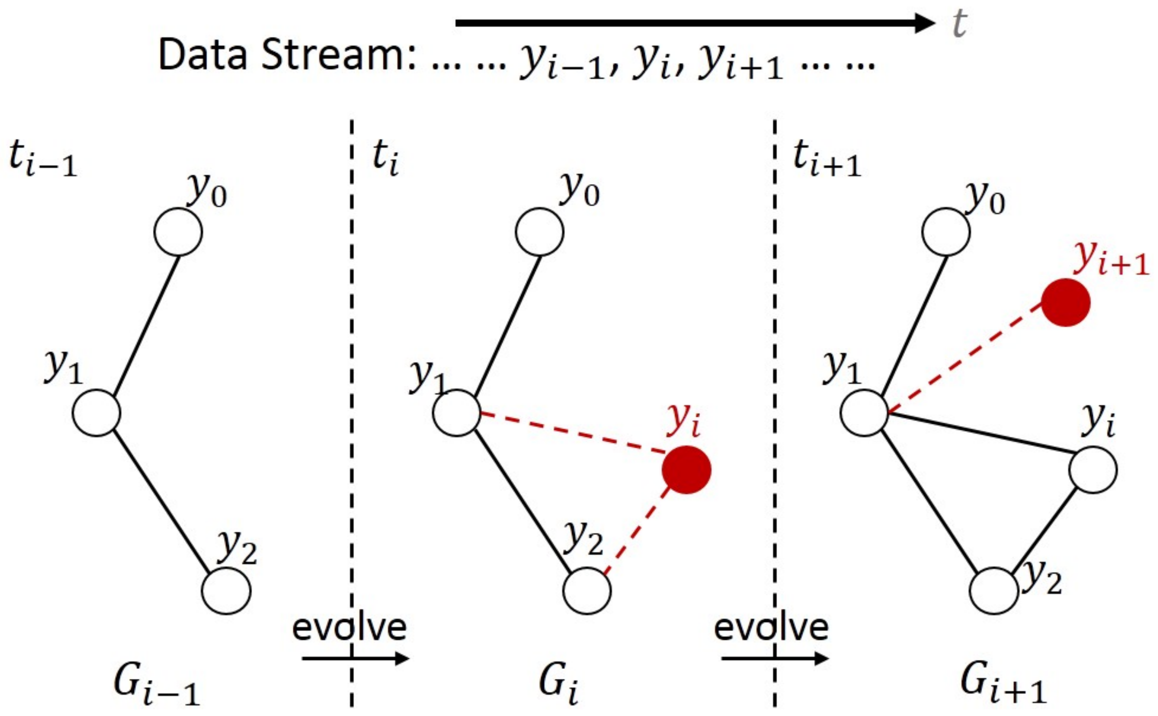
  Related work: ***active learning for networked data*** [23, 25, 3, 4, 10, 27, 8, 22]

- A novel problem: ***active learning for streaming networked data***

  To deal with both challenges 1 & 2.

Problem Formulation

Streaming Networked Data



When a new instances y_i arrives, new edges are added to connect the new instance and existing instances.

Problem Formulation

Notations for Streaming Networked Data

Let $\Delta = \{\delta_i\}_{i=0}^{\infty}$ denote a data stream and each datum be denoted as a 4-tuple

$$\delta_i = (\mathbf{x}_i, t_i, \Upsilon_i, y_i)$$

$\mathbf{x}_i$	A data instance, represented as a feature vector.
t_i	The time when the instance arrives in the data stream.
Υ_i	A set of undirected edges connected to earlier arrived instances.
y_i	An associated label in {+1, -1} (we consider binary classification problem in this paper) to represent the category of the instance.

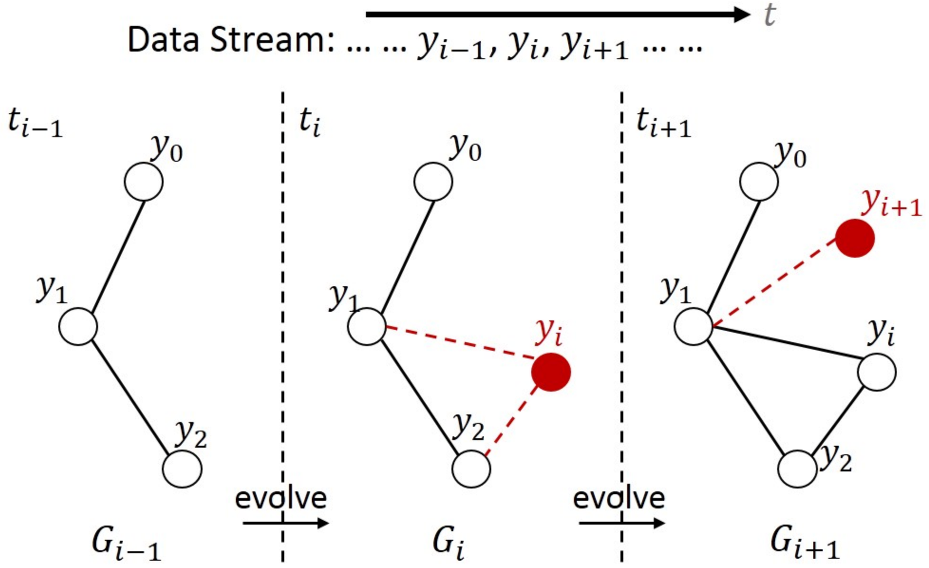
Problem Formulation

Active Learning for Streaming Networked Data

Our output is a data stream $\Delta = \{\delta_i\}_{i=0}^{\infty}$. At any time, we maintain a classifier $\mathcal{C}_i$ based on arrived instances.

At any time t_i, we go through the following steps:

1. Predict the label for $\mathbf{x}_i$ based on $\mathcal{C}_{i-1}$
2. Decide whether to query for the true label y_i
3. Update the model to be $\mathcal{C}_i$



Our goal is to use a small number of queries, to control (minimize) the accumulative error rate.

Challenges

Challenges

Concept drift.

The distribution of input data and network structure change over time as we are handling streaming data. How to adapt to concept drift?

Network correlation.

In the networked data, there is correlation among instances. How to model the correlation in the streaming data?

Online query.

We must decide whether to query an instance at the time of its appearance, which makes it infeasible to optimize a global objective function. How to develop online query algorithms?

Modeling Networked Data

Time-Dependent Network

At any time t_i, we can construct a time-dependent network G_i based on all the arrived instances before and at time t_i.

$$G_i = (\mathbf{X}_i, E_i, \mathbf{y}_i^L, \mathbf{y}_i^U)$$

$\mathbf{X}_i$	A matrix, with an element x_{ij} indicating the j^{th} feature of instance $\mathbf{x}_i$
E_i	The set of all edges between instances.
$\mathbf{y}_i^L$	A set of labels of instances that we have already actively queried before.
$\mathbf{y}_i^U$	A set of unknown labels for all the other instances.

Modeling Networked Data

The Basic Model: Markov Random Field

Given the graph G_i , we can write the energy as

$$Q_{G_i}(\boxed{\overline{\mathbf{y}}_i^L}, \mathbf{y}_i^U; \boldsymbol{\theta}) = \sum_{y_j \in \overline{\mathbf{y}}_i^L \cup \mathbf{y}_i^U} \boxed{f(\mathbf{x}_j, y_j, \boldsymbol{\lambda})} + \sum_{e_l \in E_i} \boxed{g(e_l, \boldsymbol{\beta})}$$

True labels of queried instances

The energy defined for instance $\mathbf{x}_i$

The energy associated with the edge $e_l = (y_j, y_k, c_l)$

Modeling Networked Data

Model Inference

We try to assign labels to $\mathbf{y}_i^U$ such that we can minimize the following energy

$$\min_{\mathbf{y}_i^U} Q_{G_i}(\overline{\mathbf{y}}_i^L, \mathbf{y}_i^U; \theta)$$

Usually intractable to directly solve the above problem.
Apply dual decomposition [17] to decompose the original problems into a set of tractable subproblems. The dual optimization problem is as follows:

$$L_{G_i} = \max_{\sigma} \sum_{e_l} \min_{\boxed{\mathbf{y}_l^U | \overline{\mathbf{y}}_l^L}} \left(g(e_l, \beta) + \boxed{\sigma_j^l(y_j)} + \boxed{\sigma_k^l(y_k)} \right)$$

Local optimization

Dual variables

Subject to $\boxed{\sum_{e_l \in \mathcal{I}_j^{t_i}} \sigma_j^l(\cdot) = f(\mathbf{x}_j, \cdot, \lambda)}$ **Global constraint**

We can solve the above objective function with projected subgradient [13].

Modeling Networked Data

Model Learning

Applying max margin learning paradigm, the objective function for parameter learning is written as

$$\min_\theta \frac{1}{2}\|\theta\|^2 + \mu\xi_\theta$$

where

$$\boxed{\xi_\theta} = \max_{\mathbf{y}_i^L,\mathbf{y}_i^U}\left\{\boxed{Q_{G_i}(\overline{\mathbf{y}}_i^L,\mathbf{y}_i^U;\theta) - Q_{G_i}(\mathbf{y}_i^L,\mathbf{y}_i^U;\theta)} + \boxed{D_y(\overline{\mathbf{y}}_i,\mathbf{y}_i)}\right\}$$

A slack variable

The margin between two configurations

Dissimilarity measure between two configurations

Modeling Networked Data

Model Learning

Applying dual decomposition, we have the dual optimization objective function as follows:

$$\begin{aligned} L_{\boldsymbol{\theta}} = \min_{\boldsymbol{\eta},\boldsymbol{\gamma}} \sum_{e_l} \max_{\mathbf{y}_l^U, \mathbf{y}_l^L | \bar{\mathbf{y}}_l^L} \Big(& g(\bar{e}_l, \boldsymbol{\beta}) + \eta_j^l(\bar{y}_j) + \eta_k^l(\bar{y}_k) \\ & - g(e_l, \boldsymbol{\beta}) - \eta_j^l(y_j) - \eta_k^l(y_k) \\ & + d_e(\bar{y}_j, \bar{y}_k, y_j, y_k) + \gamma_j^l(y_j) + \gamma_k^l(y_k) \Big) \end{aligned}$$

$$\text{s.t.} \quad \sum_{e_l \in \mathcal{I}_j^{t_i}} \boxed{\eta_j^l(\cdot)} = f(\mathbf{x}_j, \cdot, \boldsymbol{\lambda}); \qquad \sum_{e_l \in \mathcal{I}_j^{t_i}} \boxed{\gamma_j^l(\cdot)} = d_v(\bar{y}_j, \cdot)$$

Dual variables **Dual variables**

The optimization problem becomes $\min_{\theta} \frac{1}{2} \| \theta \|^2 + \mu L_{\theta}$

We can solve the above problem with projected subgradient method.

Streaming Active Query

Structural Variability

Intuition: control the gap between the energy of the inferred configuration and that of any other possible configuration.

We define the structural variability as follows:

$$\mathcal{V}_{\theta}^{i}(\mathbf{y}_i^L) = \max_{\mathbf{y}_i^U}\left(\boxed{Q_{G_i}(\bar{\mathbf{y}}_i^L, \mathbf{y}_i^U; \theta)} - \boxed{Q_{G_i}(\bar{\mathbf{y}}_i^L, \hat{\mathbf{y}}_i^U; \theta)}\right)$$

The energy of any other configuration

The energy of the inferred configuration

Streaming Active Query

Properties of Structural Variability

1. Monotonicity. Suppose $\mathbf{y}_1^L$ and $\mathbf{y}_2^L$ are two sets of instance labels. Given $\boldsymbol{\theta}$, if $\mathbf{y}_1^L \subsetneq \mathbf{y}_2^L$, then we have

$$\mathcal{V}_\theta^i(\mathbf{y}_1^L) \geq \mathcal{V}_\theta^i(\mathbf{y}_2^L)$$

The structural variability will not increase as we label more instances in the MRF.

2. Normality. If $\mathbf{y}_i^U = \varnothing$, we have

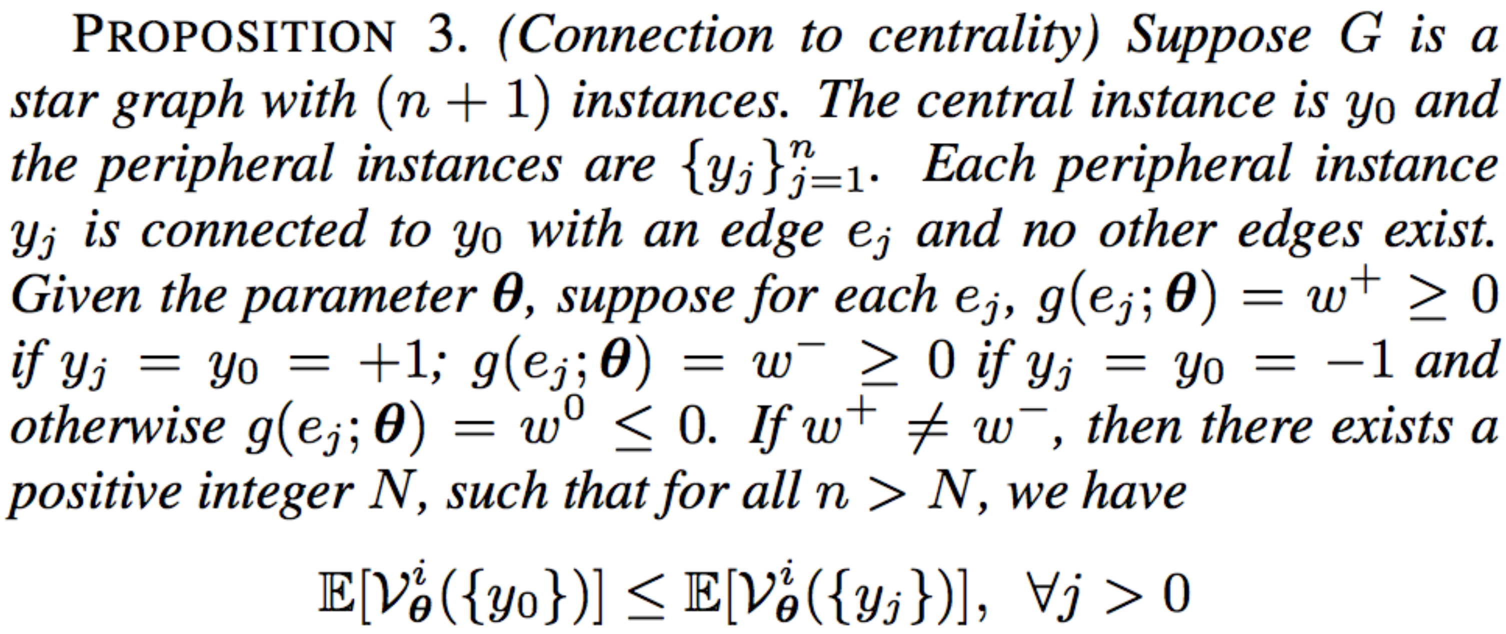
$$\mathcal{V}_\theta^i(\mathbf{y}_i^L) = 0$$

If we label all instances in the graph, we incur no structural variability at all.

Streaming Active Query

Properties of Structural Variability

3. Centrality



PROPOSITION 3. *(Connection to centrality) Suppose G is a star graph with $(n+1)$ instances. The central instance is y_0 and the peripheral instances are $\{y_j\}_{j=1}^{n}$. Each peripheral instance y_j is connected to y_0 with an edge e_j and no other edges exist. Given the parameter $\boldsymbol{\theta}$, suppose for each e_j, $g(e_j;\boldsymbol{\theta}) = w^+ \geq 0$ if $y_j = y_0 = +1$; $g(e_j;\boldsymbol{\theta}) = w^- \geq 0$ if $y_j = y_0 = -1$ and otherwise $g(e_j;\boldsymbol{\theta}) = w^0 \leq 0$. If $w^+ \neq w^-$, then there exists a positive integer N, such that for all $n > N$, we have*

$$\mathbb{E}[\mathcal{V}_{\boldsymbol{\theta}}^{i}(\{y_0\})] \leq \mathbb{E}[\mathcal{V}_{\boldsymbol{\theta}}^{i}(\{y_j\})], \ \forall j > 0$$

Under certain circumstances, minimizing structural variability leads to querying instances with high network centrality.

Streaming Active Query

Decrease Function

We define a decrease function for each instance y_i

$$\Phi^i = \boxed{\mathcal{V}_\theta^i(\mathbf{y}_{i-1}^Q)} - \boxed{\mathcal{V}_\theta^i(\mathbf{y}_{i-1}^Q \cup \{y_i\})}$$

Structural variability before querying y_i

Structural variability after querying y_i

The second term is in general intractable. We estimate the second term by expectation

$$\hat{\mathcal{V}}_{\boldsymbol{\theta}}^i = \sum_{y \in \mathcal{Y}} \boxed{P^*(\bar{y}_i = y)} \mathcal{V}_{\boldsymbol{\theta}}^i(\mathbf{y}_{i-1}^L \cup \{y_i = y\})$$

The true probability

We approximate the true probability by

$$P(\bar{y}_i = y) = \frac{e^{-Q_y^i}}{e^{-Q_y^i} + e^{-Q_{-y}^i}}$$

Streaming Active Query

Decrease Function

We define a decrease function for each instance y_i

$$\Phi^i = \boxed{\mathcal{V}_\theta^i(\mathbf{y}_{i-1}^Q)} - \boxed{\mathcal{V}_\theta^i(\mathbf{y}_{i-1}^Q \cup \{y_i\})}$$

Structural variability
before querying y_i

Structural variability
after querying y_i

The first term can be computed by dual decomposition. The dual problem is

$$L_{\boldsymbol{\theta}} = \min_{\mathbf{x}} \sum_{e_l} \max_{\mathbf{y}_l^U | \bar{\mathbf{y}}_l^L, \hat{\mathbf{y}}_l^U} \Big(g(e_l, \boldsymbol{\beta}) + \chi_j^l(y_j) + \chi_k^l(y_k)$$

$$-g(\bar{e}_l, \boldsymbol{\beta}) - \chi_j^l(\bar{y}_j) - \chi_k^l(\bar{y}_k) \Big)$$

$$\text{s.t.} \quad \sum_{e_l \in \mathcal{I}_j^{t_i}} \chi_j^l(\cdot) = f(\mathbf{x}_j, \cdot, \boldsymbol{\lambda})$$

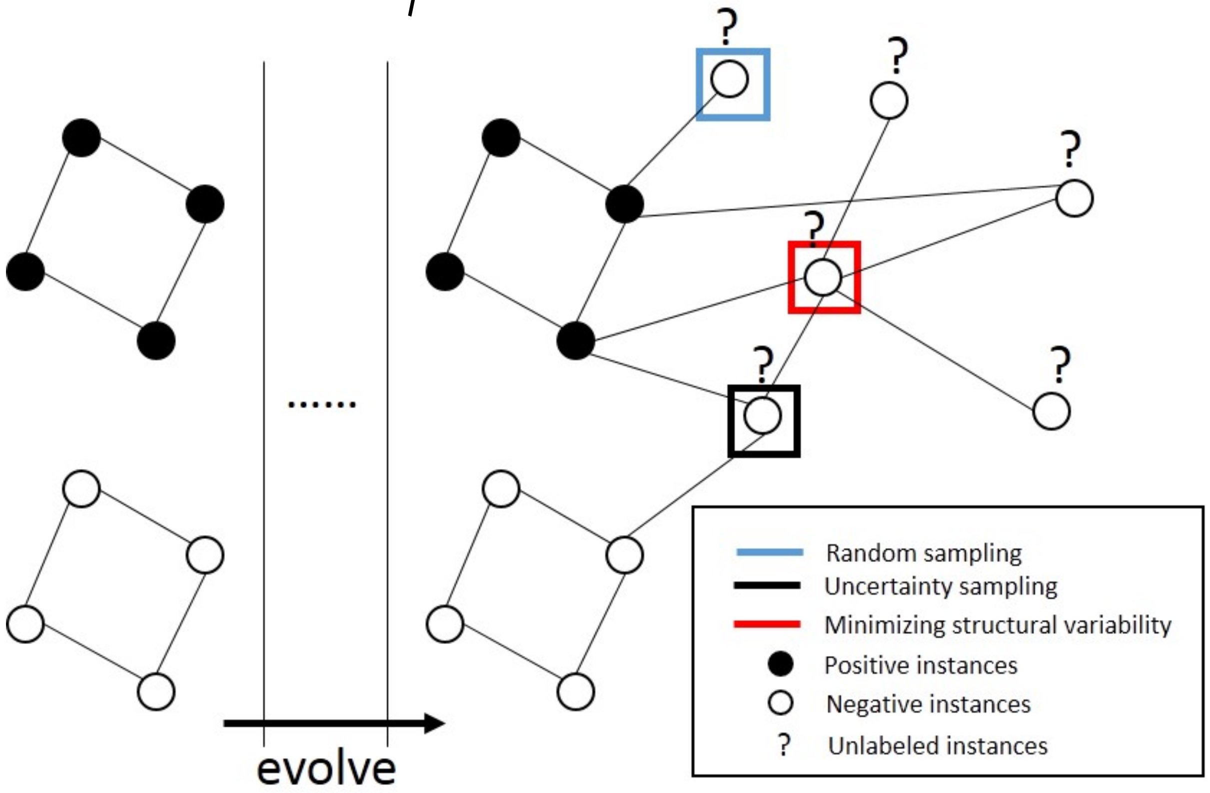
Streaming Active Query

The algorithm

Given the constant threshold κ, we query y_i if and only if

$$\phi^i \geq \kappa$$

Analysis



Enhancement by Network Sampling

Basic Idea

Maintain an instance reservoir of a fixed size, and update the reservoir sequentially on the arrival of streaming data.

Which instances to discard when the size of the reservoir is exceeded?

Simply discard early-arrived instances may deteriorate the network correlation. Instead, we consider the loss of discarding an instance in two dimensions:

1. **Spatial dimension**: the loss in a snapshot graph based on network correlation deterioration
2. **Temporal dimension**: integrating the spatial loss over time

Enhancement by Network Sampling

Spatial Dimension

Use dual variables as indicators of network correlation.

The violation for instance can be written as

$$\Gamma_{G_i}(y_k) = f(\mathbf{x}_k, y_k, \boldsymbol{\lambda}) - \sum_{e_l \in \mathcal{I}_k^{t_i}} \sigma_k^l(y_k) \longrightarrow$$

Measure how much the optimization constraint is violated after remove the instance

Then the spatial loss is

$$\Lambda_{t_i}(y_j) = \sum_{y_k \in N_j^{t_i}} \Gamma_{G_i \setminus y_j}(y_k) = \sum_{y_k \in e_l \in \mathcal{I}_j^{t_i}} \sigma_k^l(y_k)$$

Intuition

1. Dual variables can be viewed as the *message* sent from the edge factor to each instance
2. The more serious the optimization constraint is violated, the more we need to adjust the dual variables

Enhancement by Network Sampling

Temporal Dimension

The streaming network is **evolving dynamically**, we should not only consider the current spatial loss.

To proceed, we assume that for a given instance y_j, dual variables of its neighbors $\sigma_k^l(y_k)$ have a distribution with an expectation μ_j and that the dual variables are independent.

We obtain an unbiased estimator for μ_j

$$\hat{\mu}_j = \sum_{y_k \in N_j^{t_i}} \sigma_k^l(y_k) / \left|\mathcal{I}_j^{t_i}\right|$$

Integrating the spatial loss over time, we obtain

$$\mathrm{Loss}_{G_i}(y_j) = \mathbb{E}\left[\int_{t_i}^{t_j+T_m} \Lambda_t(y_j)dt\right]$$

Suppose edges are added according to preferential attachment [2], the loss function is written as

$$\mathrm{Loss}_{G_i}(y_j) = C\Lambda_{t_i}(y_j)\left((t_j+T_m)^{\frac{3}{2}} - t_i^{\frac{3}{2}}\right)$$

Enhancement by Network Sampling

The algorithm

At time t_i, we receive a new datum from the data stream, and update the graph.
If the number of instances exceed the reservoir size, we **remove the instance with the least loss function** and its associated edges from the MRF model.

Interpretation

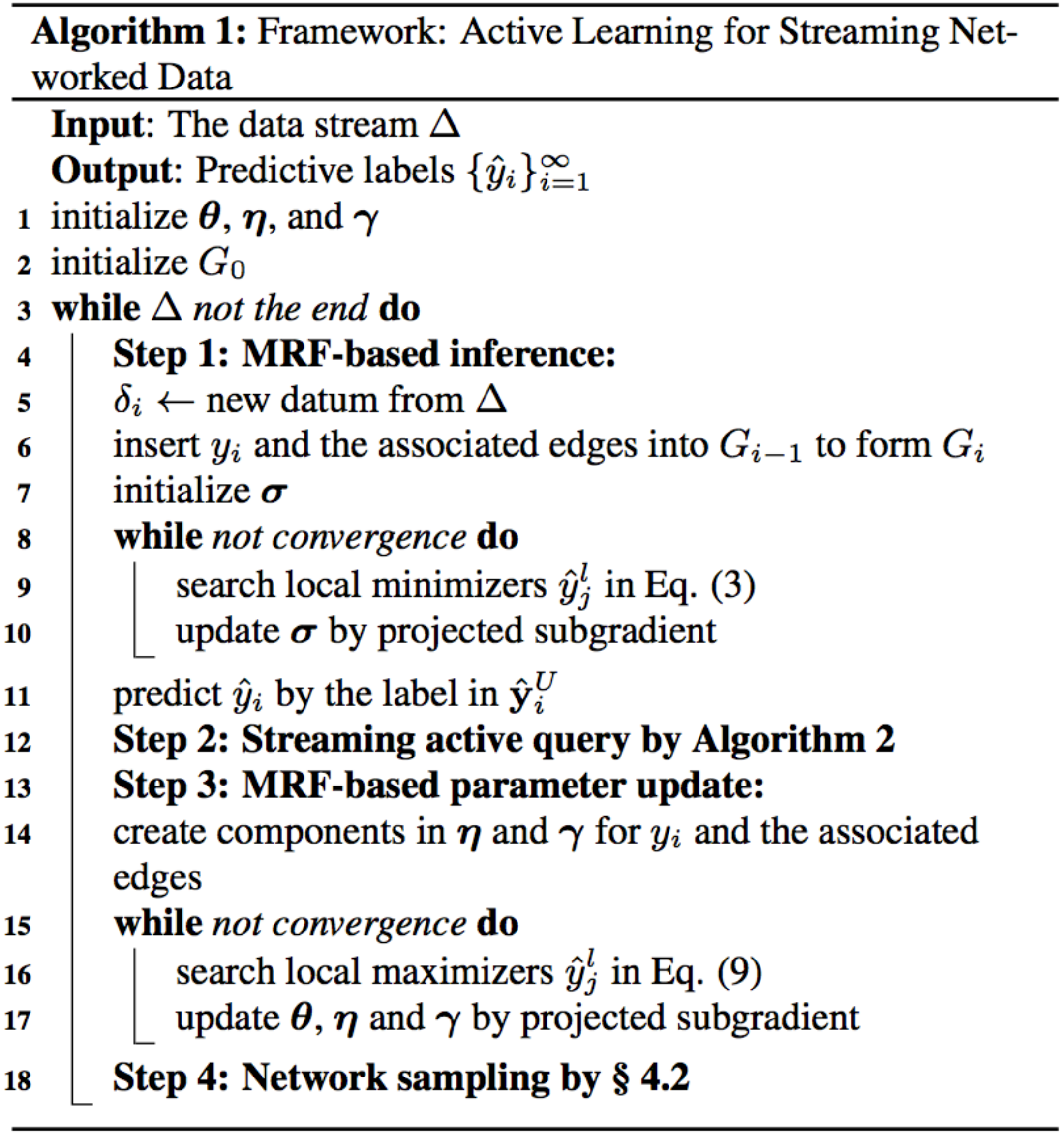
The first term $\Lambda_{t_i}(y_j)$

- Enables us to leverage the spatial loss function in the network.
- Instances that are important to the current model are also likely to remain important in the successive time stamps.

The second term $\left((t_j + T_m)^{\frac{3}{2}} - t_i^{\frac{3}{2}}\right)$

- Instances with larger t_j are reserved.
- Our sampling procedure implicitly handled concept drift, because later-arrived instances are more relevant to the current concept [28].

The Framework



Algorithm 1: Framework: Active Learning for Streaming Networked Data

```
Input: The data stream $\Delta$
Output: Predictive labels $\{\hat{y}_i\}_{i=1}^{\infty}$
initialize $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\gamma}$
initialize $G_0$
while $\Delta$ not the end do
    Step 1: MRF-based inference:
    $\delta_i \leftarrow$ new datum from $\Delta$
    insert $y_i$ and the associated edges into $G_{i-1}$ to form $G_i$
    initialize $\boldsymbol{\sigma}$
    while not convergence do
        search local minimizers $\hat{y}_j^l$ in Eq. (3)
        update $\boldsymbol{\sigma}$ by projected subgradient
    predict $\hat{y}_i$ by the label in $\hat{\mathbf{y}}_i^U$
    Step 2: Streaming active query by Algorithm 2
    Step 3: MRF-based parameter update:
    create components in $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ for $y_i$ and the associated edges
    while not convergence do
        search local maximizers $\hat{y}_j^l$ in Eq. (9)
        update $\boldsymbol{\theta}$, $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ by projected subgradient
    Step 4: Network sampling by § 4.2
```

Step 1: MRF-based inference

Step 2: Streaming active query

Step 3: MRF-based parameter update

Step 4: Network sampling

Experiments

Datasets

- **Weibo** [26] is the most popular microblogging service in China.
  - View the retweeting flow as a data stream.
  - Predict whether a user will retweet a microblog.
  - 3 types of edge factors: friends; sharing the same user; sharing the same tweet
- **Slashdot** is an online social network for sharing technology related news.
  - Treat each follow relationship as an instance.
  - Predict "friends" or "foes".
  - 3 types of edge factors: appearing in the same post; sharing the same follower; sharing the same followee.
- **IMDB** is an online database of information related to movies and TVs.
  - Each movie is treated as an instance.
  - Classify movies into categories such as *romance* and *animation*.
  - Edges indicate common-star relationships.
- **ArnetMiner** [19] is an academic social network.
  - Each publication is treated as an instance.
  - Classify publications into categories such as *machine learning* and *data mining*.
  - Edges indicate co-author relationships.

Experiments

Datasets

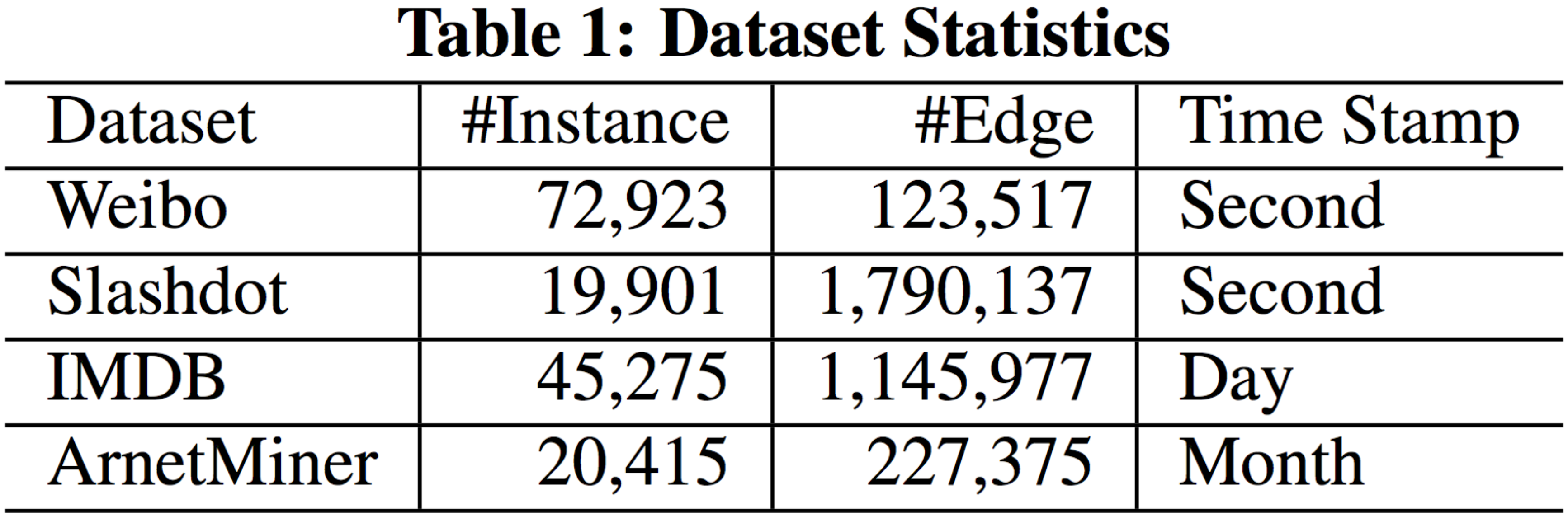
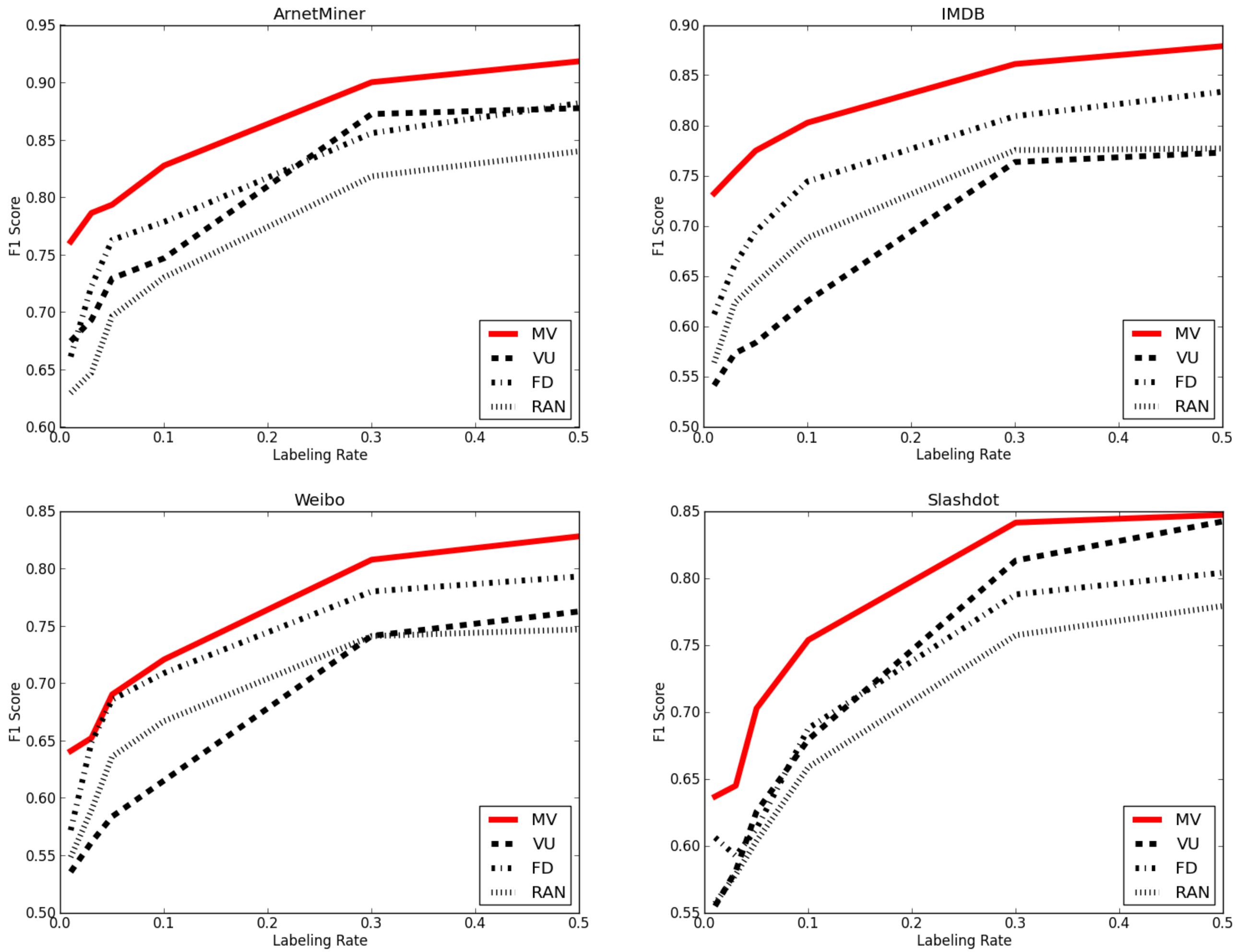


Table 1: Dataset Statistics

Dataset	#Instance	#Edge	Time Stamp
Weibo	72,923	123,517	Second
Slashdot	19,901	1,790,137	Second
IMDB	45,275	1,145,977	Day
ArnetMiner	20,415	227,375	Month

Experiments

Active Query Performance



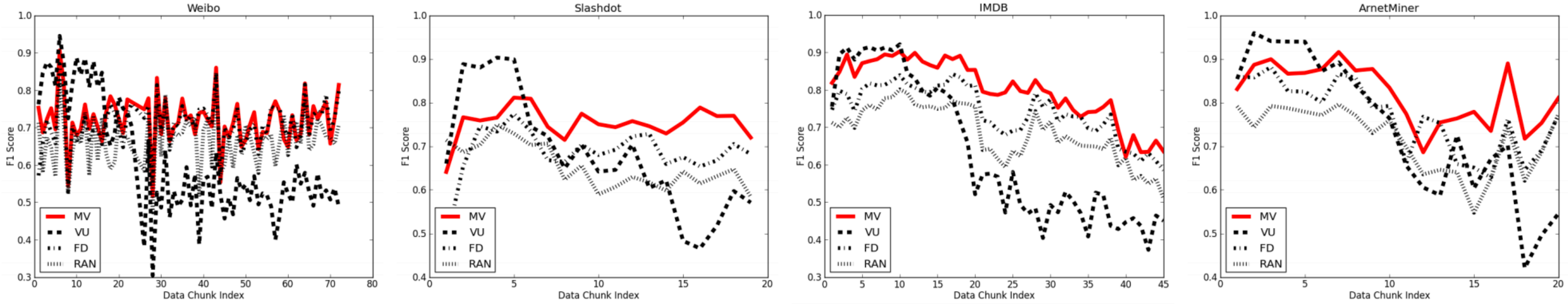
Suppress the network sampling method by setting the reservoir size to be infinite.

Compare different streaming active query algorithms.

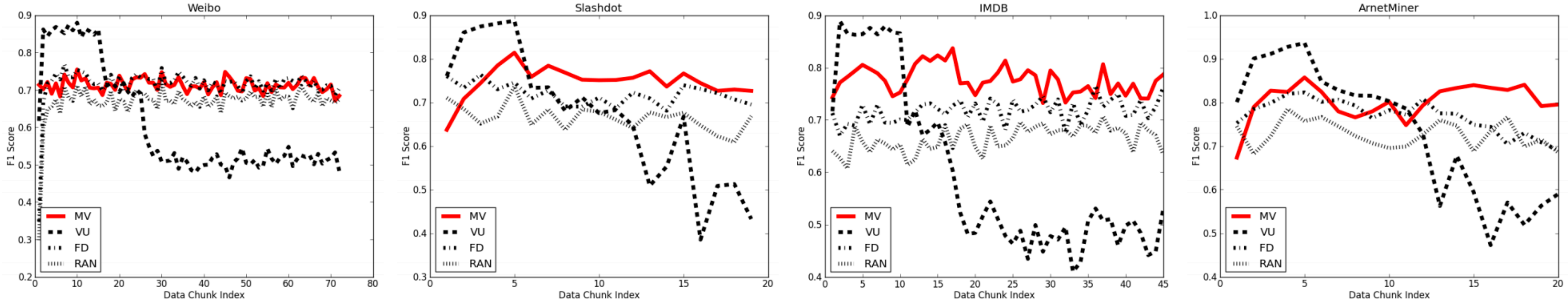
(F1 score v.s. labeling rate)

Experiments

Concept Drift



(a) Weibo (b) Slashdot (c) IMDB (d) ArnetMiner



(e) Weibo (f) Slashdot (g) IMDB (h) ArnetMiner

First row: data stream

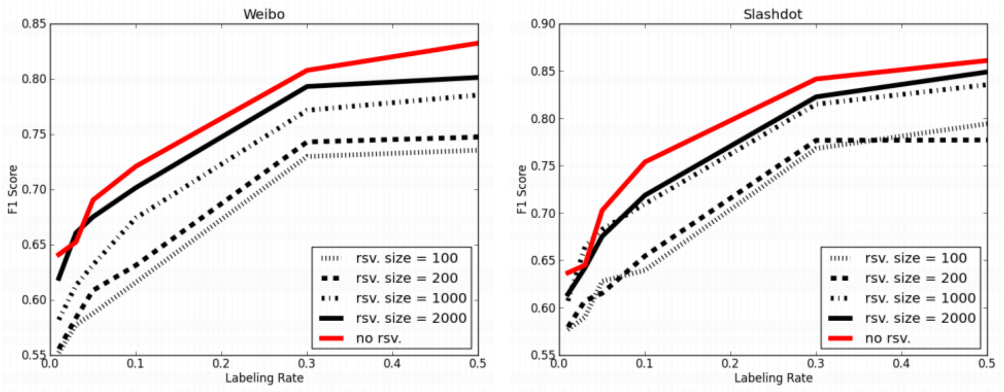
Second row: shuffled data

(F1 score v.s. data chunk index)

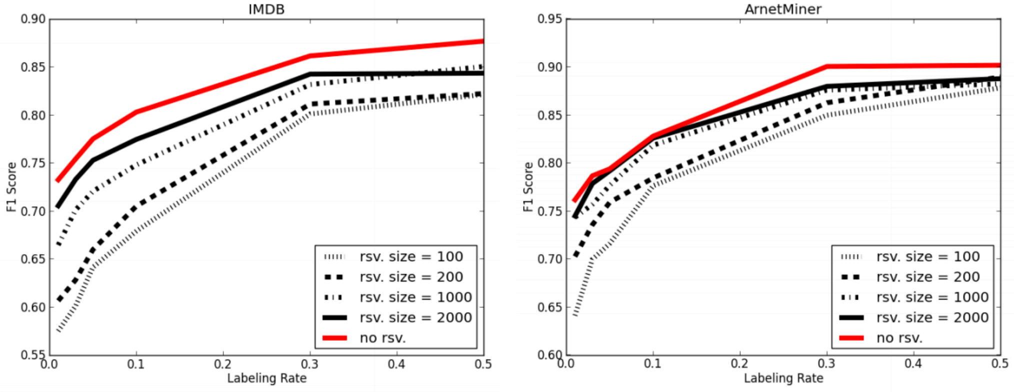
1. **Clearly found some evidence about the existence of concept drift**
2. **Our algorithm is robust because it not only better adapts to concept drift (upper row) but also performs well without concept drift (lower row).**

Experiments

Streaming Network Sampling

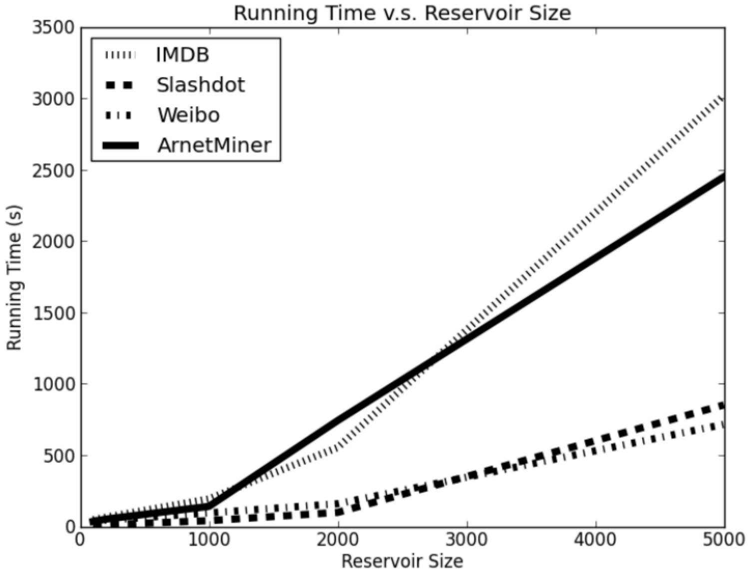


(a) Weibo (b) Slashdot



(c) IMDB (d) ArnetMiner

F1 v.s. labeling rate (with varied reservoir size)

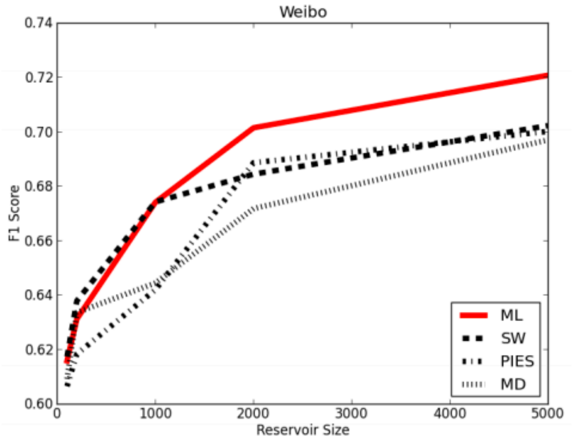


Speedup Performance (Running time v.s. reservoir size)

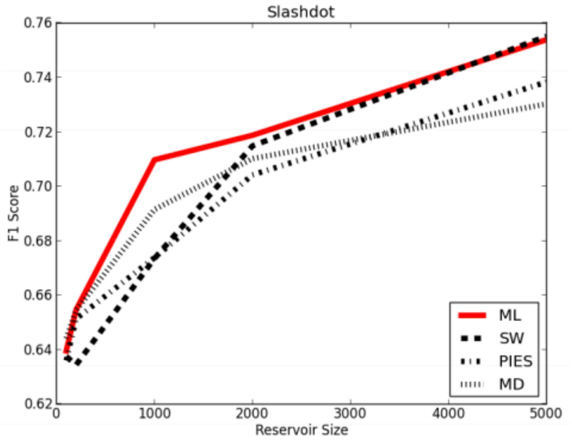
The decrease of the reservoir size leads to minor decrease in performance but significantly less running time.

Experiments

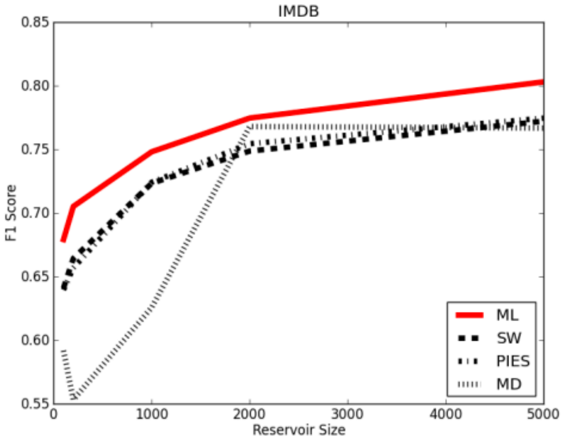
Streaming Network Sampling



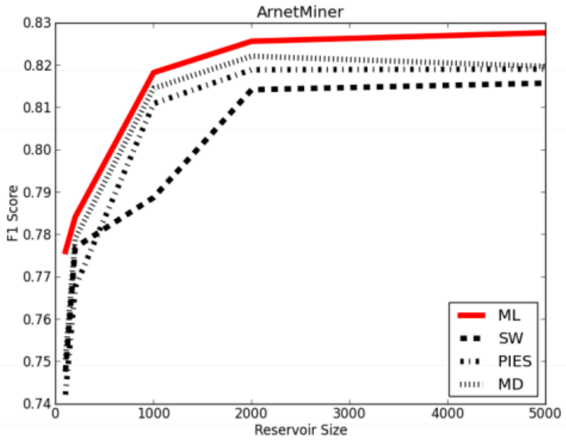
(a) Weibo



(b) Slashdot



(c) IMDB



(d) ArnetMiner

We fix the labeling rate, and compare different streaming network sampling algorithms with varied reservoir sizes.

Experiments

Performance of Hybrid Approach

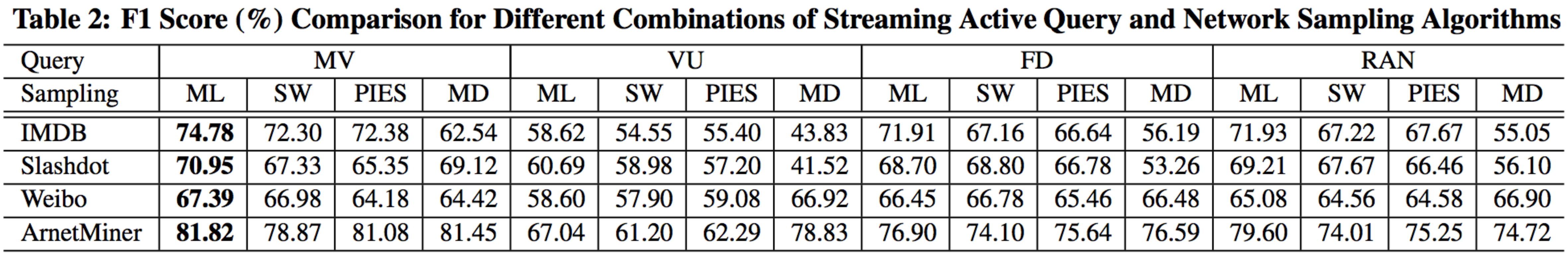


Table 2: F1 Score (%) Comparison for Different Combinations of Streaming Active Query and Network Sampling Algorithms

Query	MV				VU				FD				RAN			
Sampling	ML	SW	PIES	MD	ML	SW	PIES	MD	ML	SW	PIES	MD	ML	SW	PIES	MD
IMDB	**74.78**	72.30	72.38	62.54	58.62	54.55	55.40	43.83	71.91	67.16	66.64	56.19	71.93	67.22	67.67	55.05
Slashdot	**70.95**	67.33	65.35	69.12	60.69	58.98	57.20	41.52	68.70	68.80	66.78	53.26	69.21	67.67	66.46	56.10
Weibo	**67.39**	66.98	64.18	64.42	58.60	57.90	59.08	66.92	66.45	66.78	65.46	66.48	65.08	64.56	64.58	66.90
ArnetMiner	**81.82**	78.87	81.08	81.45	67.04	61.20	62.29	78.83	76.90	74.10	75.64	76.59	79.60	74.01	75.25	74.72

We fix the labeling rate and reservoir size, and compare different combinations of active query algorithms and network sampling algorithms.

Conclusions

- Formulate a novel problem of active learning for streaming networked data
- Propose a streaming active query algorithm based on the structural variability
- Design a network sampling algorithm to handle large volume of streaming data
- Empirically evaluate the effectiveness and efficiency of our algorithm

Thanks

Zhilin Yang, Jie Tang, Yutao Zhang

Computer Science Department, Tsinghua University