Charu C. Aggarwal
IBM T J Watson Research Center
Hawthorne, NY
USA

On Classification of Graph Streams

SIAM Conference on Data Mining, 2011

Introduction

- Massive graph streams are created by underlying activity in a number of network applications

- Examples include:
  - Communication Networks
  - Social Networks
  - Web Applications

- Present algorithms for graph stream classification

Streaming Model

- Our model assumes a stream of graph objects
- Each object is labeled with a class
- Each object contains a set of nodes and edges from the same base domain

Examples

- A bibliographic object from the DBLP network may be expressed as a graph with nodes corresponding to authors, conference, or topic area.

- A movie object from IMDB can be represented as an entity-relation graph, with edges corresponding to relationships between different elements.

- Events in social networks may lead to local patterns of activity, which may be modeled as streams of graph objects.

- The user browsing pattern at a web site is a stream of graph objects.

  - Edges $\Rightarrow$ Path taken by the user across the different objects.

Challenging Assumptions

- Stream scenario creates constraints on algorithmic design.

- The *number of distinct edges* is extremely large.

  - A graph with more than 10^8 nodes may contain as many as 10^{15} distinct (potential) edges.

  - Hard to store even summary information about distinct edges or subgraphs.

- **Additional Challenge:** The edges of a given object may occur out-of-order.

  - Creates challenges for algorithms, which extract structural characteristics for graphs, because all edges of a graph object may not be available at a given time.

Notations and Definitions

- Denote node set by N (very large)

- The individual graphs in the stream are denoted by $G_1 \ldots G_n \ldots$.

- Each graph G_i is associated with the class label C_i which is drawn from $\{1 \ldots m\}$.

- The edges of each graph G_i may not be neatly received at a given moment in time $\Rightarrow$ May appear *out of order* in the data stream.
  - The edges are received as $< EdgeId, GraphId >$

Classification Modeling Approach

- Design a rule-based classifier which relates subgraph patterns to classes
  - Left hand side contains the subgraph and right hand side contains the class-label
- Rules are maintained *indirectly* in the form of a continuously updatable and stream-friendly data structure.
- Use two criteria to mine subgraphs for rule-generation:
  - **Relative Presence:** Determine subgraphs for which relative presence of co-occurring edges (as a group) is high.
  - **Class Distribution:** Determine subgraphs which are discriminative towards a particular class.

Modeling Relative Presence of Subgraphs

- Determine subgraphs which have significant presence in terms of the *relative frequency* of its constituent edges.

- $f_\cap(P) \Rightarrow$ Fraction of graphs in $G_1 \ldots G_n$ in which **all** edges of subgraph P are present.

- $f_\cup(P) \Rightarrow$ Fraction of graphs in which **at least one or more** of the edges of subgraph P are present.

- The **edge coherence** $C(P)$ of the subgraph P is denoted by $f_\cap(P)/f_\cup(P)$.

Observations

- The definition of edge coherence is focussed on *relative presence* of subgraph patterns rather than the absolute presence.
  - This ensures that only significant patterns are found.
  - Ensures that large numbers of irrelevant patterns with high frequency but low significance are not considered.
- Computationally more challenging than direct support-based computation.

Class Confidence

- Among all graphs containing subgraph P, determine the fraction belonging to class label r
  - Also referred to as **confidence** of pattern P with respect to the class r.

- The dominant class confidence $DI(P)$ or subgraph P is defined as the maximum class confidence across all the different classes $\{1 \ldots m\}$.

- A significantly large value of $DI(P)$ for a particular test instance indicates that the pattern P is very relevant to classification.

Formal Definition (Significant Patterns)

- A subgraph P is said to be be (α, θ)-significant, if it satisfies the following two *edge-coherence* and *class discrimination* constraints:
  - The edge-coherence $C(P)$ of subgraph P is at least α.

$$C(P) \geq \alpha \tag{1}$$

  - The dominant class confidence $DI(P)$ is at least θ.

$$DI(P) \geq \theta \tag{2}$$

Broad Approach

- **Aim:** Design a continuously updatable synopsis data structure, which can be efficiently mined for the most discriminative subgraphs.

- Small size synopsis:
  - Can be dynamically maintained and applied in online fashion at any point during stream progression.
  - The *structural synopsis* maintains sufficient information which is necessary for classification purposes.

Probabilistic Synopsis

- We describe a probabilistic *min-hash approach* for determining discriminative subgraphs.

- Technique has been used earlier for dense subgraph mining applications.
  - Cannot be easily adapted to this scenario because of the large number of distinct edges and stream assumption.

- We use a 2-dimensional compression technique in which a min-hash function will be used in combination with a more straightforward randomized hashing technique.

Two Phase Description

- The min-hashing scheme corresponds to row-compression and straightforward hashing corresponds to column compression

- First describe compression using rows only
  - Subsequently describe how to add column compression to the scheme

- Sequential description eases explanation of approach

Min-hash Approach

- Coherence probability for edge set P is $f_{\cap}(P)/f_{\cup}(P)$
  - Can be estimated by sampling rows in the $GraphIds \times Edges$ matrix

- Use random sort order on the rows and examine the first row which contains at least one 1-bit in the columns for P.
  - Sorting approach is simply a way of randomly sampling *relevant* rows $\Rightarrow$ Those which have at least one 1-bit for columns of P
  - What fraction of samples have all 1-bits for P, if repeated random sorts are used?

Min-hash Approach

- Simulate the sort by using a random-hash function on the row-identifiers, and keep track of *first* (or *minimum hash value*) row index for which the corresponding bit is 1 in each column.

- Check if minimum hash index is same across all columns of set $P \Rightarrow$ Probability same as Jaccard Coefficient (or coherence probability $C(P)$)

- Repeat approach with k independent hash functions $\Rightarrow$ Compute fraction of k samples for which the minimum hash-index of the k columns of P are the same.

- **Key:** Create a data structure of minimum hash indices.

Dynamic Maintenance

- Store running minimum hash values and indices *for each column*.

- For each incoming edge, we generate k random hash values, and compare to current minimum value for that column.

- Update the running min-hash index (row index) and value if the min-hash value is lower.

- For a problem with L distinct edges, this creates a data structure of size $k \times L$

Creating Transaction Set from Min-hash sample

- For each row, determine the column identifiers for which the min-hash indices are the same.

- Create a set of transactions $\mathcal{T}$, such that each transaction contains the set of column identifiers for which the min-hash indices are the same.

- **Claim:** The coherence probability $C(P)$ of an edge set P can be estimated as the absolute support of that set in the transaction set $\mathcal{T}$, divided by k.

Columnwise Compression

- Min-hash size of $k \times L$ is still quite large, if number of distinct edges L are large

- Apply an additional layer of compression by applying a hash-function to the different columns.

- The hash function maps all columns to the range $[1, n] \Rightarrow$ Apply same approach after mapping

  – Creates a many-to-one mapping between original and compressed column set

  – Improves space efficiency at the expense of reduced accuracy

  – Accuracy reduction is modest, if average size a of stream graphs is much less than n ($a << n$)

Determining Discriminative Patterns

- Keep track of the class labels during the min-hashing scheme.

- Assume that class labels of the graphs are appended to the identifier $Id(G)$ for each graph G.

- Note that the global distribution of class labels in the min-hash summary *may not be the same* as the original data stream, because of its inherent bias in representing graph identifiers with larger number of edges in the summary transaction set $\mathcal{T}$.

- How do we estimate class confidences?

Observation

- For a particular pattern containing a *fixed number of edges*, the following is true:
  - The class fraction for any particular pattern P and class computed over the transaction set $\mathcal{T}$ is an unbiased estimate of its true value.

Classification Approach

- Approach can use synopsis structure to classify a graph at any time during the computation process.

- Determine the patterns relevant to a particular test instance.

- Pick highest frequency class among the first r relevant sub-graphs with highest dominant confidence.

Accuracy of Approach (Row Compression/Coherence Probability)

- First estimate accuracy of min-hash portion (without column compression).

- The probability of a pattern P determined from $\mathcal{T}$ to be a false positive (based on coherence probability), when using a coherence threshold of $\alpha \cdot (1 + \gamma)$ and k samples is given by at most $e^{-\alpha \cdot k \cdot \gamma^2/3}$, where e is the base of the natural logarithm.

- The number of samples k required in order to guarantee a probability at most δ for any of the determined patterns to be a false positive is given by $3 \cdot \ln(1/\delta)/(\alpha \cdot \gamma^2)$.

Accuracy of Approach (Column Compression)

- Let $f'_{\cup}(P)$ be the estimated support of P on the column-compressed data with the use of a uniform hash functions. Then, the expected value of $f'_{\cup}(P)$ satisfies the following relationship:

$$f_{\cup}(P) \leq E[f'_{\cup}(P)] \leq f_{\cup}(P) + \frac{a \cdot |P|}{n} \tag{3}$$

- Let $f'_{\cap}(P)$ be the estimated support of P on the column-compressed data with the use of a uniform hash functions. Then, the expected value of $f'_{\cap}(P)$ approximately satisfies the following relationship:

$$f_{\cap}(P) \leq E[f'_{\cap}(P)] \leq f_{\cap}(P) + \frac{a \cdot |P|}{n} \tag{4}$$

Accuracy of Approach (Class Discrimination)

- The probability of a pattern P determined from $\mathcal{T}$ to be a false positive (based on class-confidence), when using a dominant confidence threshold of $\theta \cdot (1 + \gamma)$ and k samples for the min-hash approach is given by at most $e^{-\alpha \cdot \theta \cdot k \cdot \gamma^2 / 3}$, where e is the base of the natural logarithm.

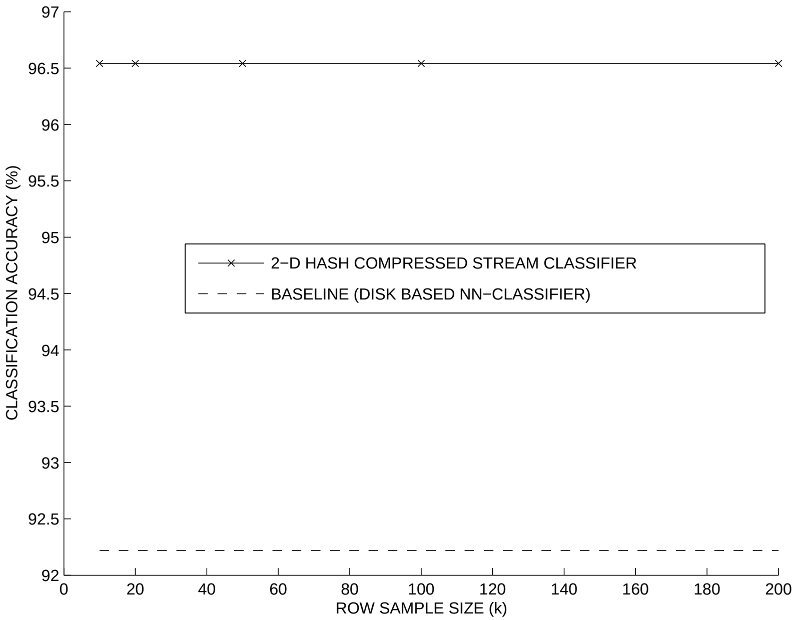
- The number of samples k required in order to guarantee a probability at most δ for any of the determined patterns to be a false positive (based on dominant class confidence) is given by $3 \cdot \ln(1/\delta)/(\alpha \cdot \theta \cdot \gamma^2)$.

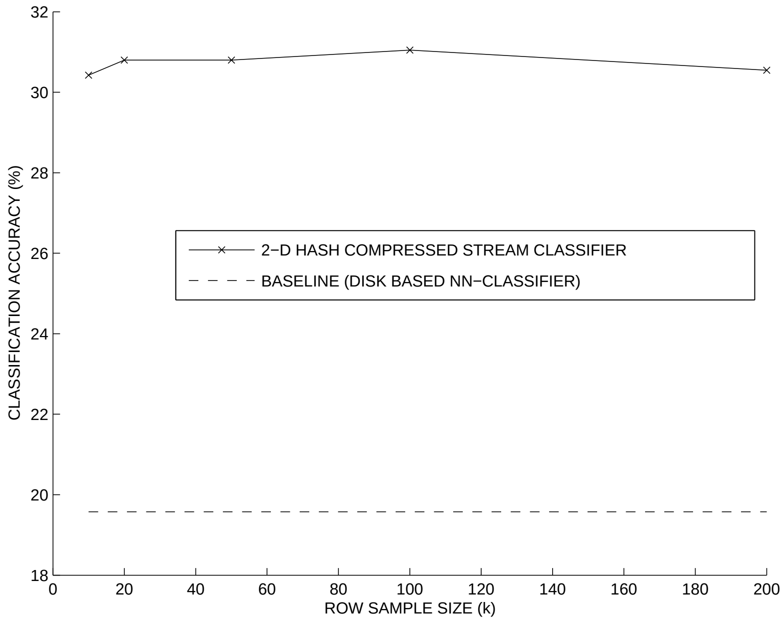
Experimental Results

- Tested on real data sets
  - DBLP and IBM Sensor Stream data set

- Compared against a disk-based baseline NN classifier
  - Accuracy of technique.
  - Efficiency of technique.
  - Sensitivity over a wide variety of parameters.

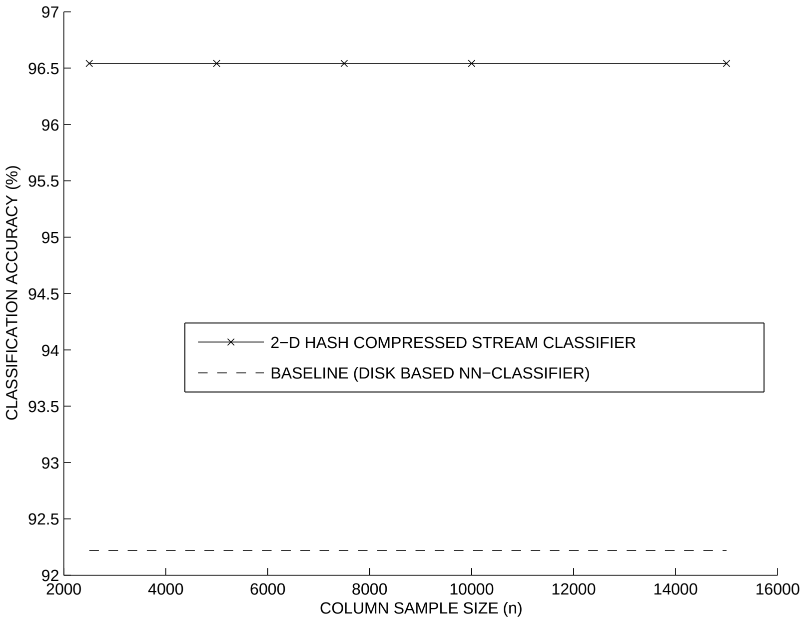
Classification Accuracy Results

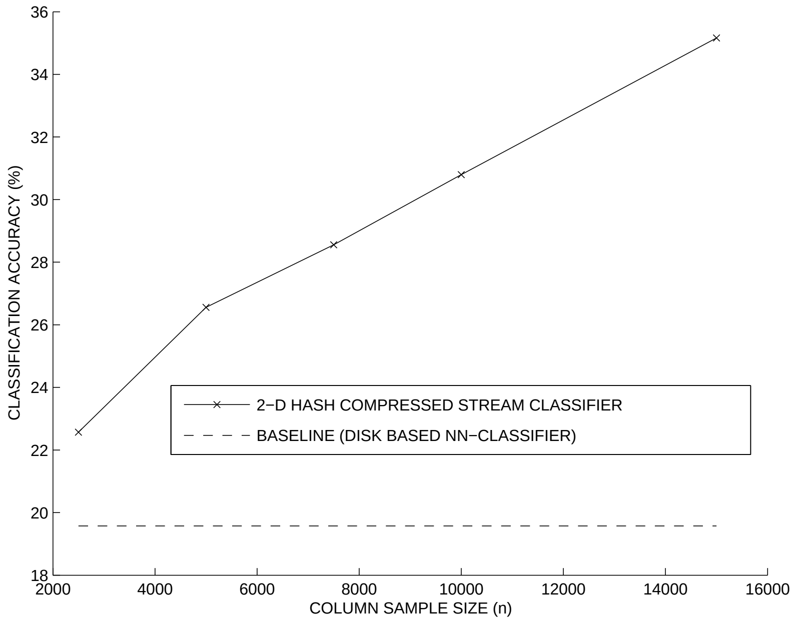




- Classification Accuracy with increasing min-hash size for (a) DBLP data set (b) Sensor data set

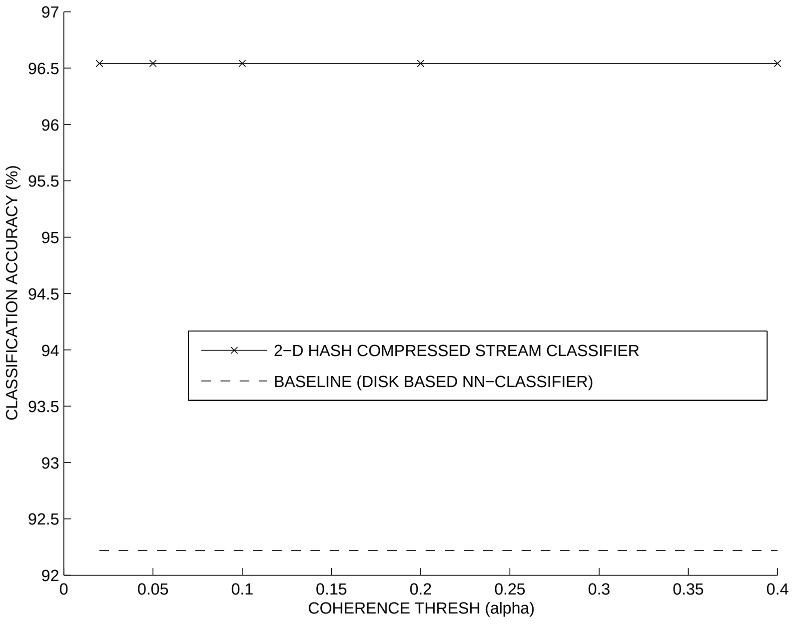
Classification Accuracy Results

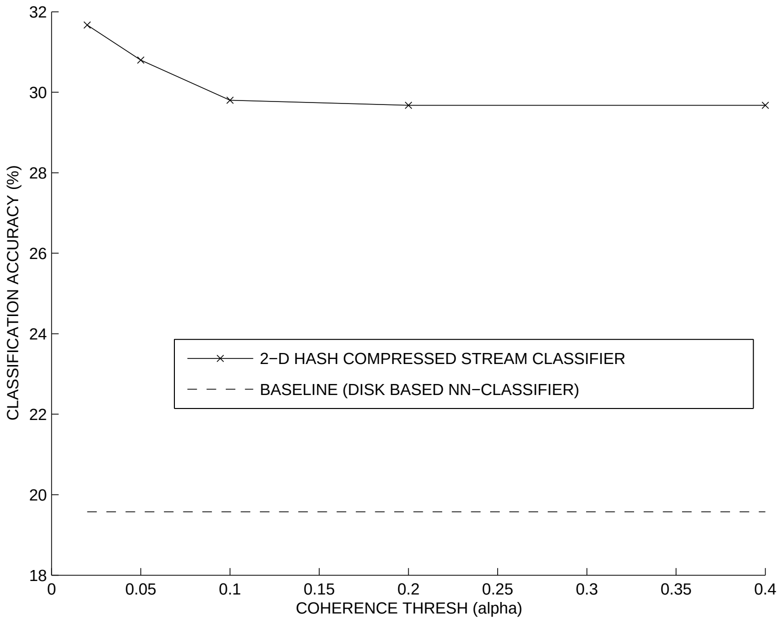




- Classification Accuracy with increasing column sample size for (a) DBLP data set (b) Sensor data set

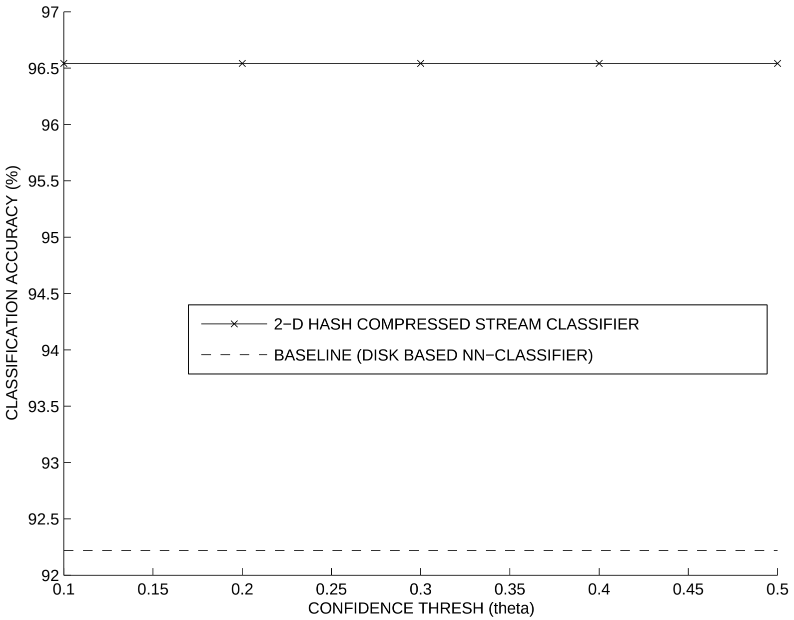
Classification Accuracy Results

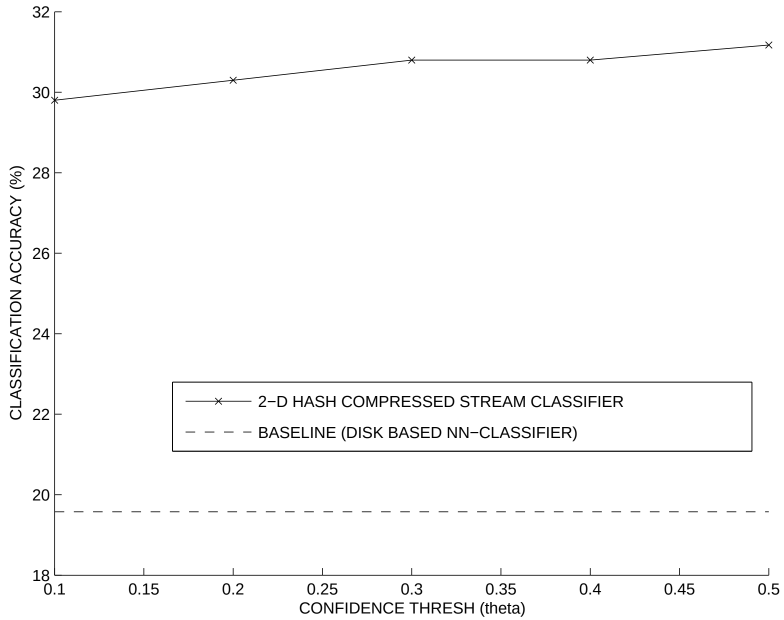




- Classification Accuracy with increasing coherence parameter for (a) DBLP data set (b) Sensor data set

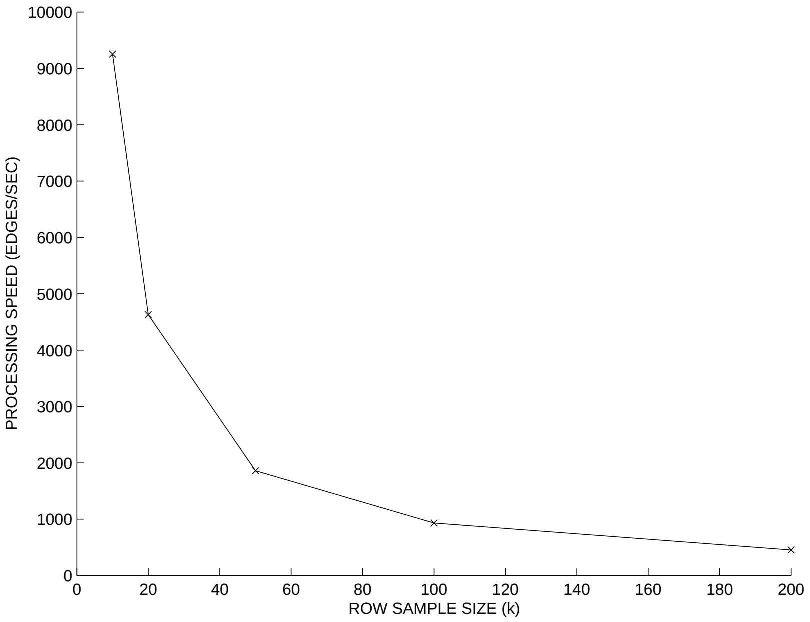
Classification Accuracy Results

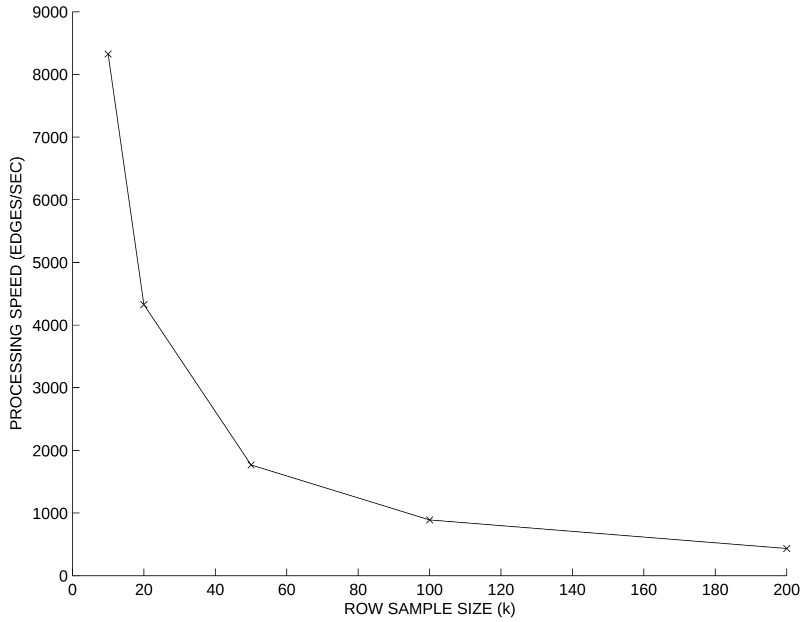




- Classification Accuracy with increasing class discrimination parameter for (a) DBLP data set (b) Sensor data set

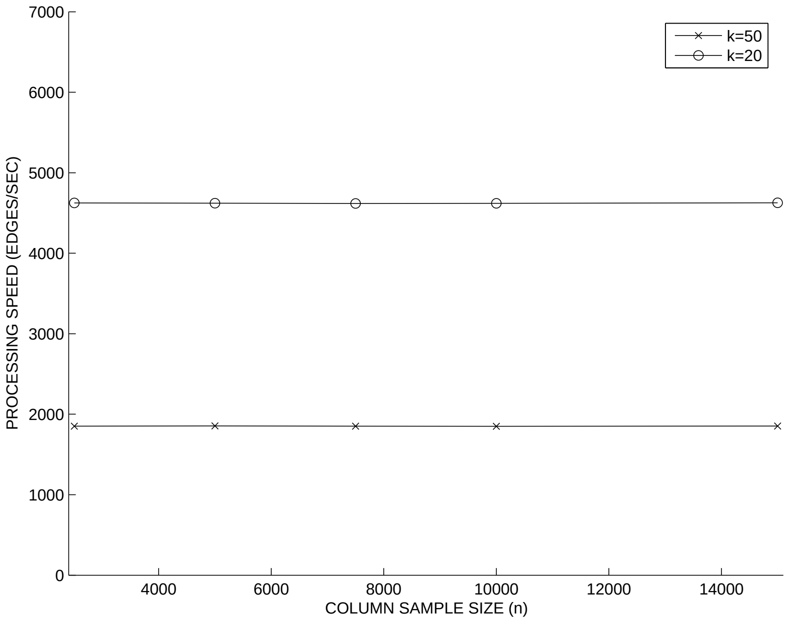
Efficiency Results

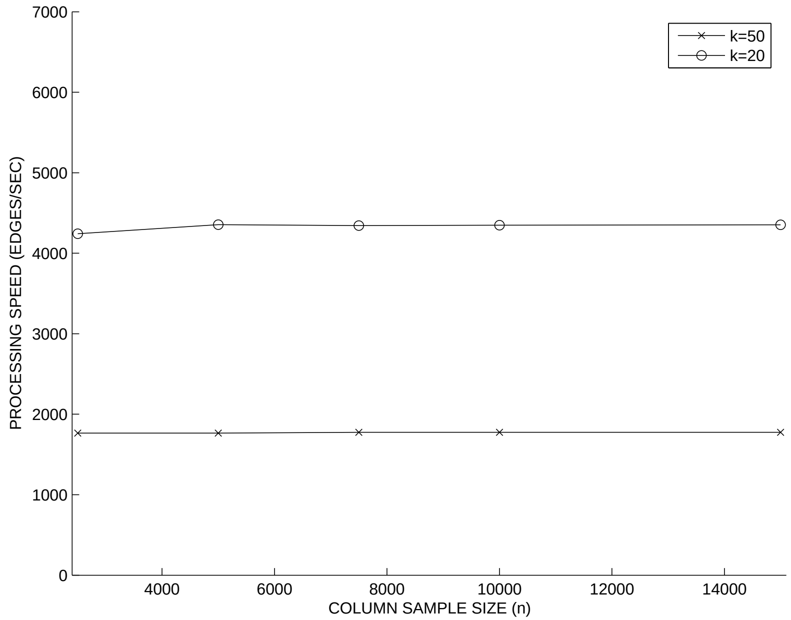




- Efficiency with increasing row compression size for (a) DBLP data set (b) Sensor data set

Efficiency Results





- Efficiency with increasing column compression size for (a) DBLP data set (b) Sensor data set

Conclusions and Summary

- New method for classification of graph streams.

- Capable of handling graph streams which are drawn from massive domains.

- Provides more effective results than a disk-based NN classifier, while maintaining efficiency.