Big Stream Data Analytics: Current & Future Trends

Latifur Khan
Professor, Department of Computer Science
The University of Texas at Dallas
www.utdallas.edu/~lkhan

This material is based upon work supported by
Agenda

- Data Streams
- Challenges
- Shortcomings of Current Solutions
- Dynamic Chunk Management
- Limited Labeled Learning
- Experiments
- Applications
- Future Direction
Data Streams

➢ Data Stream:
  – is continuous flow of data.
  – very common in today’s connected digital world.
  – important source of knowledge that enables to take extremely important decisions in (near) real time.

➢ Hence, data stream mining is very important.
Data Stream Classification

- Uses past data to build classification model.
- Predicts the labels of future instances using the model.
- Helps decision making.

Network traffic, Firewall

Expert analysis and labeling

Model update

Block and quarantine

Classification model

Benign traffic

Attack traffic
Challenge: Infinite Length

- Impractical to store and use all historical data
  - requires infinite storage
  - and running time
Challenge: Concept Drift

A data chunk

- Negative instance •
- Positive instance ○

Instances victim of concept-drift •
Challenge: Concept Evolution

Classification rules:

R1. if \((x > x_1 \text{ and } y < y_2)\) or \((x < x_1 \text{ and } y < y_1)\) then class = +

R2. if \((x > x_1 \text{ and } y > y_2)\) or \((x < x_1 \text{ and } y > y_1)\) then class = -

Existing classification models misclassify novel class instances
Existing Techniques: Ensemble based Approaches

Masud et al. [1][2]

![Diagram of ensemble learning approach]

---

[1] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani M. Thuraisingham: A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data. ICDM 2008: 929-934

[2] Mohammad M. Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen, Nikunj C. Oza: Facing the reality of data stream classification: coping with scarcity of labeled data. Knowl. Inf. Syst. 33(1): 213-244 (2011)
Existing Techniques: Ensemble Techniques

- Divide the data stream into equal sized chunks
  - Train a classifier from each data chunk
  - Keep the best $t$ such classifier-ensemble
  - Example: for $t = 3$

Note: $D_i$ may contain data points from different classes
Novel Class Detection

Masud et al. [1][2], Khateeb et al. [3]

- Non parametric
  - does not assume any underlying model of existing classes

- Steps:
  1. Creating and saving decision boundary during training
  2. Detecting and filtering outliers
  3. Measuring cohesion and separation among test and training instances

[1] Mohammad M. Masud, Qing Chen, Latifur Khan, Charu C. Aggarwal, Jing Gao, Jiawei Han, Ashok N. Srivastava, Nikunj C. Oza: Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams. IEEE Trans. Knowl. Data Eng. 25(7): 1484-1497 (2013)

[2] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani M. Thuraisingham: Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints. IEEE Trans. Knowl. Data Eng. 23(6): 859-874 (2011)

[3] Tahseen Al-Khateeb, Mohammad M. Masud, Latifur Khan, Charu C. Aggarwal, Jiawei Han, Bhavani M. Thuraisingham: Stream Classification with Recurring and Novel Class Detection Using Class-Based Ensemble. ICDM 2012: 31-40
Training with Semi-Supervised Clustering

Legend:
- Black dots: unlabeled instances
- Colored dots: labeled instances

Impurity based Clustering
Semi Supervised Clustering

Masud et al. [1][2]

 Assertion objective function (dual minimization problem)

\[ \mathcal{O}_{MCIKmeans} = \sum_{i=1}^{K} \left( \sum_{x \in \mathcal{X}_i} ||x - u_i||^2 \right) + \sum_{x \in \mathcal{L}_i} ||x - u_i||^2 \cdot Imp_i \]

**Intra-cluster dispersion**  
**Cluster impurity**

\[ Imp_i = \text{Aggregated dissimilarity count}_i \cdot \text{Entropy}_i = ADC_i \cdot \text{Ent}_i \]

Aggregated dissimilarity count (ADC):

\[ ADC_i = \sum_{x \in \mathcal{L}_i} DC(x, y) \]

\[ DC(x, y) = \begin{cases} 0 & \text{if } x \text{ is unlabeled (i.e., } y = \phi) \\ |\mathcal{L}_i| - |\mathcal{L}_i(c)| & \text{if } x \text{ is labeled and its label } y=c \end{cases} \]

Entropy (Ent):

\[ Ent_i = \sum_{c=1}^{C} (-p^i_c \cdot \log(p^i_c)) \]

The minimization problem is solved using the Expectation-Maximization (E-M) framework.

[1] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani M. Thuraisingham: A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data. ICDM 2008: 929-934

[2] Mohammad M. Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen, Nikunj C. Oza: Facing the reality of data stream classification: coping with scarcity of labeled data. Knowl. Inf. Syst. 33(1): 213-244 (2011)
Outlier Detection and Filtering

Test instance inside decision boundary (not outlier)

Test instance outside decision boundary

Raw outlier or Routlier

Test instance inside decision boundary

Routlier

Ensemble of L models

M1

M2

... 

Mt

Routlier

Routlier

Routlier

AND

x

X is an existing class instance

True

False

X is a filtered outlier (Foutlier)
(potential novel class instance)

Foutliers may appear as a result of novel class, concept-drift, or noise. Therefore, they are filtered to reduce noise as much as possible.
Novel Class Detection

1. **(Step 1)** Test instance $x$ is fed into an ensemble of $L$ models $M_1, M_2, ..., M_L$.

2. **(Step 2)** The ensemble output is then passed through an AND gate. If $x$ is classified as **True**, it proceeds to the next step; if **False**, it is treated as an existing class instance.

3. **(Step 3)** $x$ is considered a filtered outlier (Foutlier) (potential novel class instance) and is fed to $q$-NSC with all models.

4. **(Step 4)** If $q$-NSC > 0 for all models for $q' > q$, then the $x$ is identified as a novel class and the process stops. Otherwise, $x$ is treated as an existing class instance.
Computing Cohesion & Separation

- \( a(x) = \text{mean distance from an Foutlier } x \text{ to the instances in } \lambda_{o,q}(x) \)
- \( b_{\text{min}}(x) = \text{minimum among all } b_c(x) \) (e.g. \( b_+(x) \) in figure)
- \( q\)-Neighborhood Silhouette Coefficient (\( q\)-NSC):

\[
q\text{-NSC}(x) = \frac{(b_{\text{min}}(x) - a(x))}{\max(b_{\text{min}}(x), a(x))}
\]

- If \( q\)-NSC\((x)\) is positive, it means \( x \) is closer to Foutliers than any other class.

\( \lambda_c(x) \) is the set of nearest neighbors of \( x \) belonging to class \( c \)

\( \lambda_o(x) \) is the set of nearest Foutliers of \( x \)
Detection of Concurrent Novel Classes

Masud et al. [1], Faria et al. [2]

• Challenges
  – High false positive (FP) (existing classes detected as novel) and false negative (FN) (missed novel classes) rates
  – Two or more novel classes arrive at a time

• Solutions
  – Dynamic decision boundary – based on previous mistakes
    • Inflate the decision boundary if high FP, deflate if high FN
  – Build statistical model to filter out noise data and concept drift from the outliers.
  – Multiple novel classes are detected by
    • Constructing a graph where outlier cluster is a vertex
    • Merging the vertices based on silhouette coefficient
    • Counting the number of connected components in the resultant (i.e., merged) graph

[1] Mohammad M. Masud, Qing Chen, Latifur Khan, Charu C. Aggarwal, Jing Gao, Jiawei Han, Bhavani M. Thuraisingham: Addressing Concept-Evolution in Concept-Drifting Data Streams. ICDM 2010: 929-934
Novel and Recurrence

Khateeb et al. [1]

[1] Tahseen Al-Khateeb, Mohammad M. Masud, Latifur Khan, Charu C. Aggarwal, Jiawei Han, Bhavani M. Thuraisingham: Stream Classification with Recurring and Novel Class Detection Using Class-Based Ensemble. ICDM 2012: 31-40
Challenges: Fixed Chunk Size/ Decay Rate

Masud et al. [1], Parker et al. [2], Aggarwal et al. [3], Klinkenberg[4], Cohen et al. [5]

- Fixed chunk size
  - requires *a priori* knowledge about the time-scale of change.
  - delayed reaction if the chunk size is too large.
  - unnecessary frequent training during stable period if chunk size is too small.

- Fixed decay rate
  - assigns weight to data instances based on their age.
  - decay constant must match the unknown rate of change.

[1] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani M. Thuraisingham: Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints. IEEE Trans. Knowl. Data Eng. 23(6): 859-874 (2011)
[2] Brandon Shane Parker, Latifur Khan: Detecting and Tracking Concept Class Drift and Emergence in Non-Stationary Fast Data Streams. AAAI 2015: 2908-2913
[3] Charu C. Aggarwal, Philip S. Yu: On Classification of High-Cardinality Data Streams. SDM 2010: 802-813
Challenges: Fixed Chunk Size

Concept Drifts

- Chunk size too large – Delayed reaction
- Chunk size too small – Performance issue

Correct | Wrong
--- | ---

Time
Solution: Adaptive Chunk Size

Concept Drifts

Adaptive Chunk Size

Correct

Wrong
Adaptive Chunk - Sliding Window

Gamma et al. [1], Bifet et al. [2], Harel et al. [3]

- Existing dynamic sliding window techniques
  - monitor error rate of the classifier.
  - Update classifier if starts to show bad performance.
  - fully supervised, which is not feasible in case of real-world data streams.

Adaptive Chunk - Unsupervised

Haque et al. [1][2]

Input

Prediction using Ensemble

Predicted Class

Classifier Confidence

Distribution Before

Distribution After

Update Classifier & Shrink Window

Change

Yes

No

Grow Window

Adaptive Chunk - Unsupervised

Haque et al. [1][2]

- Prediction using Ensemble
- Predicted Class
- Association
- Purity
- Model 2 Confidence
- Model t Confidence
- Classifier Confidence
- Update Classifier & Shrink Window
- Grow Window
- Change
- Yes
- No

[1] Ahsanul Haque, Latifur Khan, Michael Baron, Bhavani M. Thuraisingham, Charu C. Aggarwal: Efficient handling of concept drift and concept evolution over Stream Data. ICDE 2016: 481-492

For each testing instance $x$:

- Confidence for $i^{th}$ model, $c_i^x = h_i^x \cdot z_i$
  
  • $h_i^x = (a_i^x, p_i^x)$ is a vector of estimator values on test instance $x$.
  
  • $z_i = \text{vector containing weights of the estimators for } i^{th} \text{ model.}$

To estimate confidence of the entire ensemble, we take the average confidence of the models towards the predicted class.
Let \( h \) be the closest cluster from data instance \( x \) in model \( M_i \), confidence of \( M_i \) in classifying instance \( x \) is calculated based on the following estimators:

- **Association**: \( a_i^x = R_h - D_i(x) \), where \( R_h \) is the radius of \( h \) and \( D_i(x) \) is the distance of \( x \) from \( h \).

- **Purity**: \( p_i^x = \frac{N_m}{N_s} \), where \( N_s \) is the number of labeled instances in \( h \), and \( N_m \) is the number of instances from the majority class in \( h \).

\[ \begin{align*}
N_s &= 15, \quad N_m = 14 \\
p_i^x &= 14/15
\end{align*} \]
Big Stream Data: Current & Future

- **Stream Mining***
  - IOT Big Stream Mining—Real Time
  - Security:
    - Encrypted Stream Traffic Analysis
      - Website Fingerprinting

Application (1): Detecting Zero-day attacks

The Distribution of attacks through time

- 28 classes
- Each class has 200 data points
- Chunk size = 100

Chunk contains 1 new attack and 5 existing classes.
Results Detecting Zero-day attacks

<table>
<thead>
<tr>
<th>BiDi Packets:</th>
<th>FP%</th>
<th>FN%</th>
<th>Err%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dxminer 1</td>
<td>26.988</td>
<td>0.0</td>
<td>24.869</td>
</tr>
<tr>
<td>Dxminer + DAE features 2</td>
<td>15.635</td>
<td>42.037</td>
<td>4.396</td>
</tr>
<tr>
<td>BiDi Packets:</td>
<td>26.988</td>
<td>0.0</td>
<td>24.869</td>
</tr>
<tr>
<td>Dxminer + DAE features 2</td>
<td>15.635</td>
<td>42.037</td>
<td>4.396</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N-grams SysCalls:</th>
<th>FP%</th>
<th>FN%</th>
<th>Err%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dxminer 1</td>
<td>31.87</td>
<td>19.33</td>
<td>21.414</td>
</tr>
<tr>
<td>Dxminer + DAE features 2</td>
<td>4.761</td>
<td>46.754</td>
<td>17.66</td>
</tr>
<tr>
<td>N-grams SysCalls:</td>
<td>31.87</td>
<td>19.33</td>
<td>21.414</td>
</tr>
<tr>
<td>Dxminer + DAE features 2</td>
<td>4.761</td>
<td>46.754</td>
<td>17.66</td>
</tr>
</tbody>
</table>

- Dxminer\(^1\) = novel class detection method
- DAE\(^2\) = Denoising Autoencoders features

Spark-based Real-time Anomaly Detection: Framework (Application 2)

- **Stream Data Mining Module**

- **Experimental Result**

<table>
<thead>
<tr>
<th>Component</th>
<th>Number of parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker for emitting tuples</td>
<td>05</td>
</tr>
<tr>
<td>Worker for clustering</td>
<td>08</td>
</tr>
<tr>
<td>Worker for prediction</td>
<td>08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of data points</th>
<th>Number of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset 1</td>
<td>10,000</td>
<td>63</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>10,000</td>
<td>134</td>
</tr>
</tbody>
</table>

**Cluster Environment**

- Experimental Result
  - Dataset 1 - Performance data of Spark jobs
  - Dataset 2 - Performance data for Yahoo Cloud Service Benchmark database operation.

**Technical Approach**

Technical Approach

Cluster Environment

<table>
<thead>
<tr>
<th>Component</th>
<th>Number of parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker for emitting tuples</td>
<td>05</td>
</tr>
<tr>
<td>Worker for statistical analysis</td>
<td>08</td>
</tr>
</tbody>
</table>

Statistical Model

<table>
<thead>
<tr>
<th>Number of data point</th>
<th>Dataset 1</th>
<th>Dataset 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of windows</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>Total Number of points</td>
<td>80,000</td>
<td>80,000</td>
</tr>
</tbody>
</table>

Testing

<table>
<thead>
<tr>
<th>Method</th>
<th>TPR</th>
<th>FNR</th>
<th>TNR</th>
<th>FPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-square based Online model</td>
<td>90.00%</td>
<td>10.00%</td>
<td>98.80%</td>
<td>1.2%</td>
</tr>
<tr>
<td>Base-line offline method</td>
<td>8.24%</td>
<td>91.76%</td>
<td>99.16%</td>
<td>0.84%</td>
</tr>
</tbody>
</table>

Application (3): Encrypted Traffic Fingerprinting

Al-Naami et al. [1][2]

- Traffic Fingerprinting (TFP) is a Traffic Analysis (TA) attack that threatens web/app navigation privacy.
- TFP allows attackers to learn information about a website/app accessed by the user, by recognizing patterns in traffic.
- Examples: Website Fingerprinting

A Framework To Recommend New Political Actors With Role In Real-time (4)

- Dictionary (CAMEO) development requires
  - Human involvement
  - Not up-to-date
  - Higher Cost
  - Processing large number of articles

- Our Goal:
  - Reduce human effort and cost
  - Recommending news actor real-time
  - Update dictionary
A Framework To Recommend New

- Political with multiple alias names,
  - e.g., Barack Hussein Obama’, ‘Barack Obama’, etc.

- Role of a political actor changes over time.
  - e.g., ’Shimon Peres’ has multiple political roles in Israel

- Processing a large volume of news articles
  - demands scalable, distributed computing
A Framework To Recommend New Political Actors With Role In Real-time

- A real-time framework for recommendation
  - Possible new actors with their roles
  - Grouping actor aliases

- Frequency-based actor ranking algorithm

- A graph-based technique to recommend roles
  - A new actor
  - Existing actor whose role varies over time
  - Integrating external knowledge base (e.g., Wikipedia)

- Time window-based recommendation system.
Real-time Political Actor Detection Over Textual Political Stream


Challenges

✓ Same actor with multiple alias names
✓ Identify novel actor along with roles
✓ Existing political actor’s role changes over time
✓ Processing high volume of news articles across the world
Future Direction

- Adversarial active learning
  - Traditional algorithms are vulnerable to adversarial manipulation.
  - Instances should be selected carefully.
- Efficient online change detection
- Deep Learning Guided Stream Mining
- Multi-stream Analytics
References

- Ahsanul Haque, Latifur Khan, Michael Baron, Bhavani M. Thuraisingham, Charu C. Aggarwal: *Efficient handling of concept drift and concept evolution over Stream Data*. ICDE 2016: 481-492
- Swarup Chandra, Ahsanul Haque, Latifur Khan, Charu C. Aggarwal: *An Adaptive Framework for Multistream Classification*. CIKM 2016: 1181-1190
- Khaled Al-Naami, Swarup Chandra, Ahmad M. Mustafa, Latifur Khan, Zhiqiang Lin, Kevin W. Hamlen, Bhavani M. Thuraisingham: *Adaptive encrypted traffic fingerprinting with bi-directional dependence*. ACSAC 2016: 177-188
- Parker, B., Khan, L.: *Detecting and tracking concept class drift and emergence in non-stationary fast data streams*. In: Twenty-Ninth AAAI Conference on Artificial Intelligence. (Jan 2015).
- Tahseen Al-Khateeb, Mohammad M. Masud, Latifur Khan, Charu C. Aggarwal, Jiawei Han, Bhavani M. Thuraisingham: *Stream Classification with Recurring and Novel Class Detection Using Class-Based Ensemble*. ICDM 2012: 31-40
- Mohammad M. Masud, Qing Chen, Latifur Khan, Charu C. Aggarwal, Jing Gao, Jiawei Han, Ashok N. Srivastava, Nikunj C. Oza: *Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams*. IEEE Trans. Knowl. Data Eng. 25(7): 1484-1497 (2013)
- Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani M. Thuraisingham: *Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints*. IEEE Trans. Knowl. Data Eng. 23(6): 859-874 (2011)
- Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani M. Thuraisingham: *A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data*. ICDM 2008: 929-934
- Mohammad M. Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen, Nikunj C. Oza: *Facing the reality of data stream classification: coping with scarcity of labeled data*. Knowl. Inf. Syst. 33(1): 213-244 (2011)
- Charu C. Aggarwal, Philip S. Yu: *On Classification of High-Cardinality Data Streams*. SDM 2010: 802-813
- Wei Fan, Yi-an Huang, Haixun Wang, Philip S. Yu: *Active Mining of Data Streams*. SDM 2004: 457-461
- Xingquan Zhu, Peng Zhang, Xiaodong Lin, Yong Shi: *Active Learning from Data Streams*. ICDM 2007: 757-762