Conference on Information Technology in Asia - CITA 2015

Big Data and Stream Analytics

Geoff Holmes  University of Waikato
Acknowledgements

This work is in collaboration with:
Dale Fletcher, Peter Reutemann (part 1);
Bernhard Pfahringer, Albert Bifet, Jesse Read,
Hendrik Blockeel and Joaquin Vanschoren (parts II & III).
Outline

- Part I
  - Big Data from an ML perspective.
  - Challenges for ML.

- Part II
  - Stream Analytics
  - Issues and solutions

- Part III (if time permits)
  - Experiment Databases
  - Recent results in Experiment Databases for Big Data
Part 1 Big Data

- I will take a machine learning perspective
  - Input is a matrix of observations $X$ and an (output) target vector $Y$
  - Problem is to learn a mapping that describes the relationship between the input and the output. This mapping is termed a model.
  - We use the model on unseen observations to predict the target (key is generalisation error).

- If we let $n$ be the number of observations which each have $m$ feature values then in big data terms either $n$ or $m$ or both are large, in data volume think petabytes ($10$ to the $15$) and exabytes ($10$ to the $18$) (and possibly zettabytes, $10$ to the $21$).

- Given the above, for ML the emphasis in terms of modelling is sub-linear algorithms (for example, $n \log n$) to produce models. Additionally, we can try to reduce $n$ by sampling and/or $m$ by dimensionality reduction.
Challenges for ML and Big Data

*Machine Learning that Matters by Kiri Wagstaff*

1. A law passed or legal decision made that relies on the result of an ML analysis.

2. $100M saved through improved decision making provided by an ML system.

3. A conflict between nations averted through high-quality translation provided by an ML system.

4. A 50% reduction in cybersecurity break-ins through ML defenses.

5. A human life saved through a diagnosis or intervention recommended by an ML system.

6. Improvement of 10% in one country’s Human Development Index (HDI)
An ML application that meets Wagstaff’s challenge #2

- 2. $100M saved through improved decision making provided by an ML system.
- Application is in soil testing, determining soil fertility status.
- Knowing this can significantly increase crop yield.
- In many parts of the world actual crop yield is much lower than potential yield.
An ML application that meets Wagstaff’s challenge 2 – facts and figures

- Application saves $18M per year, every fourth year it saves $30M.
- Application has been running since 2006.
- Over 250 separate models have been deployed.
- n varies from 5K to 150K
- m varies from 1.5K to 10K
Near Infrared Spectroscopy

- Obtain samples and reference data from existing technology (e.g., wet chemistry) – establish targets Y.
- Process same samples using a proxy (e.g., NIR) – to form X
Other Instrument Data

- Can use other instruments as proxy: Mid Infrared and XRF

Such data can be fused together to form an instance for learning.
Why is this a good application area?

Correlation coefficient: 0.9706
Mean absolute error: 1.3557
Root mean squared error: 2.8067
Advanced DAita Mining System (ADAMS), Motivation

- We needed to experiment and deploy solutions with one system
- Given the need to connect to a LIMS, a “plug-on” architecture is ideal
- All data needs to be pre-processed and assessed, especially if data is fused together
What is ADAMS?

- Tree-based workflow engine
- Visual programming paradigm (loops, choice, sequence)
- Has multi-threaded branching (complex data)
- Is self-documenting
- Supports experimentation leading to application development
- Integrates with R, Weka, MOA, Twitter
Samples can be mislabelled when collected or swapped by mistake in the lab.

Incorrectly labelled samples will go through the wrong processes.

A QA step is to predict the sample type from its NIR spectrum.

What is the best pre-processing and classifier to optimise classification?
ADAMS – deploy sample type checker

- Deployed solution incorporates best pre-processing and classifier
- Polls for new incoming spectra (so dynamic)
- Checks stored sample type against predicted
- Emails those samples that differ for manual checking (and includes what it thinks the sample should be)
ADAMS – deployment features

- Data mining process is mainly about experimentation and then deployment.
- Deployment requires integration into existing dynamic systems
- Workflows need to contain components to enable all of this
- Example components for our systems include (some present in KNIME also):
  - Database access
  - Web services
  - PDF report generation
  - Email support
  - Spreadsheet support and so forth
Conclusions

- An ADAMS solution, satisfying Wagstaff’s challenge #2 has been deployed since 2006.
- Experimenting with various forms of data pre-processing is as important as finding the best classifier (in app dev)
- ADAMS helps you to both experiment and deploy solutions.
- All of the above applies to Big Data more generally – so we need Big Data Pre-Processing to be a serious area of research.
Part II – Massive Online Analysis (MOA)
What is MOA?

- {M}assive {O}nline {A}nalysis is a framework for online learning from data streams.
- It is closely related to WEKA
- It includes a collection of offline and online methods as well as tools for evaluation covering:
  - Classification
  - Clustering
- Easy to extend
- Easy to design and run experiments
Why is it needed?
Classification Setting

1. New training examples at any point in time
2. Must work in finite memory
3. Expect concept drift
4. Anytime prediction

Evaluation procedures:
- Holdout
- Interleaved Test-Then-Train or Prequential
Classifiers and Prediction Strategies

- Naïve Bayes (and NBM)
- Decision Stumps
- kNN
- Hoeffding Tree (VFDT)
- Hoeffding Option Tree
- Hoeffding Adaptive Tree
- Bagging and Boosting
- ADWIN Bagging
- Leveraging Bagging
- Adaptive Size HT
- SGD, Perceptron

- Prediction Strategies
  - Majority Class
  - Naïve Bayes Leaves
  - Adaptive Hybrid

- Explicit drift detection
  - ADWIN
  - DDM
  - EDDM
Example – real dataset

Electricity Dataset, Accuracy

Accuracy, %

Time, instances

$10^4$

VFDT  Majority Class  Naive Bayes
Magic Classifier

Electricity Dataset, Accuracy

Accuracy, %

Time, instances

0 1 2 3 4 \times 10^4

- Magic Classifier
- VFDT
- Majority Class
- Naive Bayes
Comparison from the literature

<table>
<thead>
<tr>
<th>Algorithm name</th>
<th>Acc. (%)</th>
<th>Algorithm name</th>
<th>Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDM</td>
<td>89.6*</td>
<td>Local detection</td>
<td>80.4</td>
</tr>
<tr>
<td>Learn++.CDS</td>
<td>88.5</td>
<td>Perceptron</td>
<td>79.1</td>
</tr>
<tr>
<td>KNN-SPRT</td>
<td>88.0</td>
<td>AUE2</td>
<td>77.3</td>
</tr>
<tr>
<td>GRI</td>
<td>88.0</td>
<td>ADWIN</td>
<td>76.6</td>
</tr>
<tr>
<td>FISH3</td>
<td>86.2</td>
<td>EAE</td>
<td>76.6</td>
</tr>
<tr>
<td>EDDM-IB1</td>
<td>85.7</td>
<td>Prop. method</td>
<td>76.1</td>
</tr>
<tr>
<td><strong>Magic classifier</strong></td>
<td><strong>85.3</strong></td>
<td>Cont. λ-perc.</td>
<td>74.1</td>
</tr>
<tr>
<td>ASHT</td>
<td>84.8</td>
<td>CALDS</td>
<td>72.5</td>
</tr>
<tr>
<td>bagADWIN</td>
<td>82.8</td>
<td>TA-SVM</td>
<td>68.9</td>
</tr>
<tr>
<td>DWM-NB</td>
<td>80.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* tested on a subset
Magic Classifier revealed

- Magic Classifier is: predicted label at time $t = \text{observed label at time } t-1$. (classifier is independent of feature space) – better name = no-change classifier

Kappa Plus Statistic

- $p_0$: classifier’s prequential accuracy
- $p_e$: no-change classifier’s prequential accuracy
- $\kappa^+$ statistic
  \[ \kappa^+ = \frac{p_0 - p_e}{1 - p_e} \]
- $\kappa^+ = 1$ if the classifier is always correct
- $\kappa^+ = 0$ if the predictions coincide with the correct ones as often as those of the no-change classifier
New evaluation measure

Electricity Market Dataset $\kappa^+$

![Graph showing Kappa Plus Statistic over time](image-url)
Temporally Augmented Classifier

**SWT**: meta strategy that builds meta instances by augmenting the original input attributes with the values of recent class labels from the past

\[ Pr[\text{class is } c] \equiv h(x^t, c^{t-\ell}, \ldots, c^{t-1}) \]

for the \( t \)-th test instance, where \( \ell \) is the size of the sliding window over the most recent true labels.
SWT improves base classifiers
State-of-the-art in data stream classification

**Instance-Incremental:** Update the model with new training examples as soon as they are available.

- Naive Bayes
- Hoeffding Decision Trees
- Neural Networks
- $k$-Nearest Neighbour (model based on a moving window)

**Batch-Incremental:** Collect $w$ training examples, then build a batch model with these examples (and drop an old model when memory is full), and repeat.

- Logistic Regression
- Decision Trees
- Support Vector Machines
- etc.
### Methods used for comparison

**Instance-Incremental Methods:**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>SGD</td>
<td>Stochastic Gradient Descent</td>
</tr>
<tr>
<td>HT</td>
<td>Hoeffding Trees</td>
</tr>
<tr>
<td>LB-HT</td>
<td>Leveraging Bagging Ensemble of HT with ADWIN</td>
</tr>
<tr>
<td>kNN</td>
<td>k-Nearest Neighbour</td>
</tr>
<tr>
<td>LB-kNN</td>
<td>Leveraging Bagging Ensemble of kNN with ADWIN</td>
</tr>
</tbody>
</table>

where Leveraging Bagging [Bifet et al., 2010] of 10 models with the ADWIN change detector; kNN window (batch) size \( -w \) 1000.

**Batch-Incremental Methods:**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWE-J48</td>
<td>Accuracy Weighted Ensemble with C4.5 Decision Trees</td>
</tr>
<tr>
<td>AWE-SVM</td>
<td>Accuracy Weighted Ensemble with Support Vector Machines</td>
</tr>
<tr>
<td>AWE-LR</td>
<td>Accuracy Weighted Ensemble with Logistic Regression</td>
</tr>
</tbody>
</table>

with Accuracy Weighted Ensemble (AWE-*) [Wang et al., 2003] of 10 models (batches), batch size \( -w \) of 500. All classifiers are from the WEKA/MOA frameworks with default parameters.
## Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NB</th>
<th>kNN</th>
<th>HT</th>
<th>AWE-J48</th>
<th>LB-HT</th>
<th>SGD</th>
<th>AWE-LR</th>
<th>AWE-SVM</th>
<th>LB-kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Newsgroups</td>
<td>68.1</td>
<td>94.9</td>
<td>94.3</td>
<td>94.7</td>
<td>94.4</td>
<td>94.9</td>
<td>88.4</td>
<td>95.6</td>
<td>DNF</td>
</tr>
<tr>
<td>IMDB</td>
<td>60.4</td>
<td>60.8</td>
<td>63.5</td>
<td>53.6</td>
<td>61.8</td>
<td>63.8</td>
<td>54.0</td>
<td>54.5</td>
<td>62.4</td>
</tr>
<tr>
<td>CovType</td>
<td>60.5</td>
<td>92.2</td>
<td>80.3</td>
<td>87.8</td>
<td>88.6</td>
<td>60.7</td>
<td>84.5</td>
<td>84.2</td>
<td>92.4</td>
</tr>
<tr>
<td>Electricity</td>
<td>73.4</td>
<td>78.4</td>
<td>79.2</td>
<td>75.3</td>
<td>88.8</td>
<td>57.6</td>
<td>70.5</td>
<td>68.6</td>
<td>80.8</td>
</tr>
<tr>
<td>Poker</td>
<td>59.5</td>
<td>69.3</td>
<td>76.1</td>
<td>77.9</td>
<td>95.0</td>
<td>68.9</td>
<td>60.9</td>
<td>60.4</td>
<td>70.3</td>
</tr>
<tr>
<td>CovPokElec</td>
<td>24.2</td>
<td>78.4</td>
<td>79.3</td>
<td>81.6</td>
<td>92.4</td>
<td>68.1</td>
<td>70.1</td>
<td>69.8</td>
<td>79.1</td>
</tr>
<tr>
<td>LED(50000)</td>
<td>54.0</td>
<td>63.2</td>
<td>68.7</td>
<td>72.0</td>
<td>73.2</td>
<td>11.8</td>
<td>73.0</td>
<td>72.8</td>
<td>69.8</td>
</tr>
<tr>
<td>SEA(50)</td>
<td>85.4</td>
<td>86.8</td>
<td>86.4</td>
<td>88.4</td>
<td>88.2</td>
<td>85.4</td>
<td>89.4</td>
<td>89.6</td>
<td>88.0</td>
</tr>
<tr>
<td>SEA(50000)</td>
<td>85.4</td>
<td>86.5</td>
<td>86.4</td>
<td>87.5</td>
<td>88.8</td>
<td>85.2</td>
<td>89.0</td>
<td>89.2</td>
<td>87.7</td>
</tr>
<tr>
<td>HYP(10,0.0001)</td>
<td>91.2</td>
<td>83.3</td>
<td>89.0</td>
<td>71.6</td>
<td>88.1</td>
<td>79.5</td>
<td>93.7</td>
<td>93.4</td>
<td>87.1</td>
</tr>
<tr>
<td>HYP(10,0.001)</td>
<td>70.9</td>
<td>83.3</td>
<td>78.8</td>
<td>75.8</td>
<td>84.8</td>
<td>71.1</td>
<td>91.8</td>
<td>92.0</td>
<td>86.9</td>
</tr>
<tr>
<td>RBF(0,0)</td>
<td>51.2</td>
<td>89.0</td>
<td>83.2</td>
<td>83.0</td>
<td>89.7</td>
<td>16.6</td>
<td>46.9</td>
<td>50.5</td>
<td>90.6</td>
</tr>
<tr>
<td>RBF(50,0.0001)</td>
<td>31.0</td>
<td>89.4</td>
<td>45.5</td>
<td>79.3</td>
<td>76.7</td>
<td>16.6</td>
<td>54.9</td>
<td>57.9</td>
<td>90.5</td>
</tr>
<tr>
<td>RBF(10,0.0001)</td>
<td>52.1</td>
<td>89.3</td>
<td>79.2</td>
<td>81.8</td>
<td>85.3</td>
<td>16.6</td>
<td>51.0</td>
<td>52.8</td>
<td>90.7</td>
</tr>
<tr>
<td>RBF(50,0.01)</td>
<td>29.1</td>
<td>84.0</td>
<td>32.3</td>
<td>51.0</td>
<td>55.7</td>
<td>16.6</td>
<td>46.5</td>
<td>50.4</td>
<td>82.1</td>
</tr>
<tr>
<td>RBF(10,0.01)</td>
<td>52.0</td>
<td>88.3</td>
<td>76.4</td>
<td>76.8</td>
<td>81.8</td>
<td>16.6</td>
<td>49.4</td>
<td>50.7</td>
<td>88.9</td>
</tr>
</tbody>
</table>

| Avg. Rank      | 7.44| 4.00| 3.51| 6.49    | 2.88  | 7.56| 5.31   | 4.69    | 2.69    |
| Avg. Accuracy  | 59.3| 82.3| 74.9| 77.4    | 83.3  | 51.9| 69.6   | 70.8    |         |
| Tot. Time (s)  | 260 | 18349 | 417 | 6883    | 9877  | 42  | 66757  | 5800    | 166312  |
| Tot. RAM-Hrs   | 0.01| 0.80| 0.54| 3.55    | 50.39 | 0.00| 37.83  | 3.49    | 77.90   |

(Format: Accuracy Rank); RAM-Hrs = hours with 1 GB in memory
State-of-the-art – recommendations

- If unsure then kNN is a safe bet
- A good model built from 1000 instances can be better than one built from a 1,000,000
- If your method is instance-incremental then find the concept drift
- If you have the resources then use an ensemble
- As with smaller scale ML, choose your method according to your data.
- How can we scale-up stream processing?
HADOOP in Real Time?

Hadoop

S4/Storm
Big Data Stream Mining

Machine Learning

Distributed

Batch
Hadoop
Mahout

Stream
S4, Storm

SAMOA

Non Distributed

Batch
R, WEKA, ...

Stream
MOA
SAMOA Architecture

Use S4, Storm, or other distributed stream processing platform
Use MOA, or other streaming machine learning library
Easy to extend through PACKAGES
Conclusion

- Data Stream Mining is a relatively new field so it is still finding its way – MOA is a good place to start. Distributed Stream Processing is a way to scale these methods to massive data.
- Importance of high-quality data sources cannot be over-stated
- Simple methods should never be discounted
ADAMS
https://mloss.org/software/view/425/
MOA
http://moa.cms.cms.waikato.ac.nz/
http://samoa-project.net

Questions or Comments?
Part III - Machine learning experimentation in practice

Low reproducibility
Hirsh (2008), Pedersen (2008)

Low reusability
Experiments repeated to answer any question

Low generalizability
Limited meta-learning

Low interpretability
Hand (2006): Illusion of progress
Benefits of sharing for machine learning

Reproducibility
Good science

Visibility
Algorithms appear in searches

Quick, easy analysis
Querying: Answer questions
Test hypotheses

Integration/Standardization
Data mining tools
import/export

Reuse
Save time & energy
(e.g. benchmarking)

Generalizability:
Plug into prior results: larger studies
Organization facilitates analysis

Reference
‘Map’ of known approaches
Compare to state-of-the-art
Includes negative results
Collaborative Experimentation

- Lessons from e-Sciences:
  - A formal representation language (e.g. MAGE-ML)
  - Ontologies: formal vocabulary + knowledge base (e.g. Gene Ontology)
  - A searchable repository (e.g. Microarray database)
  - Interfaces

Interface (API)

- Exposé
- ExpML
- ExpDB

DM toolboxes

- Query interface
- Mining
- Meta-models

Theoretical examination
ExpML: a markup language for DM experiments

- Free exchange (publishing) of DM experiments, XML-based

```
<expml>
  <dataset id='d1'>
    <learner_appl id='op1'>
      <learner_impl name=... version=...>
        <parameter_setting name='P' value='100'/>
      </learner_impl>
      <learner_component role='base-learner'>
        <learner_appl id='op2'>
          ...
        </learner_appl>
      </learner_component>
    </learner_appl>
    <performance_estimation_appl id='op3'>
      ...
    </performance_estimation_appl>
    <evaluation_function_appl id='op4'>
      <evaluation_list id='e1'>
        <evaluation name='accuracy' value='0.99'/>
      </evaluation_list>
    </evaluation_function_appl>
  </dataset>
  <source id='d1'>
    <sink id='e1'>
      <link source='d1' sink='op3:input_data'/>
      ...
    </sink>
  </source>
</expml>
```
Experiment database

>650,000 experiments, 54 algorithms, >87 datasets, 45 evaluation measures, 2 data processors, bias-variance analysis
Learn – gaining insights from querying the database

- Analysis can be broken down into:
  - Model-level (HOW an algorithm performs)
  - Data-level (WHEN, on which kinds of data an algorithm is expected to behave)
  - Method-level (WHY an algorithm behaves in a certain way)
Learning from the past (model-level)

How did previous algorithms perform on ‘letter’ dataset?
Learning from the past

Quick hypothesis testing

Effects of kernel width? Interplay with number or attributes?
Learning from the past (model-level)

Compare all algorithms over all UCI datasets (Caruana&Niculescu06)
Learning from the past (model-level)

Do some approaches rank significantly better on average?

[Graph showing average Friedman rank and rank scaled by critical difference for various machine learning algorithms, including Boosting, Bagging, SVM, MultiLayerPerceptron, RandomForest, AdaBoost, LogisticModelTrees, C45, LinearLogisticRegr, Ripper, NaiveBayesTree, LogisticRegression, RBFNetwork, PART, kNN, NNrules, NaiveBayes, OneR.]
Learning from the past (data-level)

Effect of dataset size and number of trees on Random Forests
Learning from the past (data-level)

Test preprocessing effects

Learning curves of various algorithms on ‘letter’ dataset?
Learning from the past (data-level)

When does J48 (C4.5) perform better than OneR?

Find meta-patterns
Learning from the past (method-level)

Bias-variance profile + effect of dataset size (Brain&Webb02)
Conclusion

- I have presented environments for
  - Developing ML applications
  - Analysing streaming algorithms
  - Analysing experimental results (generating hypotheses)
- All are Open Source
- All could be better with more Open Data
ADAMS
https://mloss.org/software/view/425/
MOA
http://moa.cms.waikato.ac.nz/
http://samoa-project.net
Expt DBS
http://openml.org/

Questions or Comments?