Data Mining Lab: Dynamic Weighted Majority for Incremental Learning

Miena Basta Badres, David Jozefiak

July 29, 2020

Overview

1. DWMIL: features and explanation
2. Imbalanced streaming data sets
3. Analysis of the “Weather” data set
4. Performance metrics
5. Results of the paper vs. reproduced results
6. Performance of DWMIL with F1-score
7. Performance of DWMIL on a new data set
8. DWMIL vs. Learn^{++}.NIE
9. Conclusion

Introduction

Concept drifts

- concept drifts occurring in data streams will jeopardize the accuracy and stability
- if the data stream is imbalanced, it will be even more challenging to detect and handle the concept drift
- these two problems have been intensively addressed separately
- they have yet to be well studied when they occur together

DWMIL
Key features

- chunk-based incremental learning method
- deals with data streams with concept drift and class imbalance problem
- creates a base classifier for each chunk
- weighs them by their performance tested on the current chunk
- a classifier trained recently or on a similar concept will receive a high weight

DWMIL

Four major merits

- can keep stable for non-drifted streams and quickly adapt to the new concept
- is totally incremental, no previous data needs to be stored
- keeps a limited number of classifiers to ensure high efficiency
- is simple and needs only one threshold parameter

DWMIL

Method explanation

- on each data chunk $\mathcal{D}(t)$ at timestamp t, a new classifier H is learned
- the new classifier H is merged with $\mathcal{H}(t-1)$ to form the set $\mathcal{H}(t)$
- classifiers are associated with the vector of weights, denoted as $w^{(t)} = [w_1^{(t)}, ..., w_m^{(t)}]^T$
- weights measure the importance of the classifiers in the set
- a weight $w_j^{(t)}$ for classifier $H_j^{(t)}$ is reduced on each timestamp
- the adjusted weight is given by $w_j^{(t)} = (1 - \epsilon_j^{(t)}) \cdot w_j^{(t-1)}$
- finally, new data x is predicted with $sign(\sum_{j=1}^{m} w_j^{(t)} \cdot H_j^{(t)}(x))$

DWMIL
UnderBagging

- combines the strength of random undersampling and bagging
- random undersampling
  - simple technique used to resolve imbalance in the data set
  - remove random samples from the majority class
  - may increase the variance of the classifier
  - may potentially discard useful or important samples
- bagging
  - special case of model averaging
  - used to improve the stability and accuracy
  - reduces variance and helps to avoid overfitting

Imbalanced streaming data sets

Real world data sets

- Weather
  - weather information of Bellevue in Nebraska
  - each day can be classified as “rainy” or “not rainy”
- Electricity
  - changes of the electricity price of New South Wales in Australia

Imbalanced streaming data sets
Synthetic data sets

- Moving Gaussian
  - consists of two Gaussian distributed classes
- SEA
  - contains three attributes ranging from 0 to 10
  - only the first two attributes are related to the class
- Hyper Plane
  - contains gradually changing decision hyperplane concepts
- Checkerboard
  - nonlinear XOR classification problem

Imbalanced streaming data sets

Further data sets

- Forest Covertype
  - contains the cover type for 30 x 30 meter forest cells
  - 581,012 instances and 54 attributes
- Poker Hand
  - consists of 1,000,000 instances
  - each instance represents a hand having five poker playing cards
  - each card is described by the attributes suit and rank

The “Weather” data set
Overview

- consists of 18,159 daily readings
  - 5,698 (31 %) are classifed as “rainy”
  - the remaining 12,461 (69 %) are classifed as “not rainy”
- missing values were synthetically generated
- 8 weather features
  - Temperature (Fahrenheit)
  - Dew Point (Fahrenheit)
  - Sea Level Pressure (hPa)
  - Visibility (Miles)
  - Average Wind Speed (Knots)
  - Maximum Sustained Wind Speed (Knots)
  - Maximum Temperature (Fahrenheit)
  - Minimum Temperature (Fahrenheit)

The “Weather” data set
Analysis

- no missing values at all
- imperial units were used, so we converted them into metric units during analysis
- all values are floats, so we can easily calculate **min**, **max**, **mean** and **std** for every value, e. g. for the temperature:
  - min = -24.3 °C
  - max = 33.6 °C
  - mean = 10.6 °C
  - std = 11.7 °C
- two major outliers in the **pressure** column (5503.8 hPa and 5503.1 hPa)

Performance metrics

F1-Score

- measure of a test's accuracy
- considers both the **Precision** and the **Recall** to compute the score

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Performance metrics

Geometric Mean Error

- Geometric Mean is the n-th root of the product of n numbers:

$$\epsilon_{gm} = 1 - \sqrt{TPR \cdot TNR}$$

- True Positive Rate (TPR) or **Recall** / **Sensitivity**:

$$TPR = \frac{TP}{TP + FN}$$

- True Negative Rate (TNR) or **Specificity**:

$$TNR = \frac{TN}{TN + FP}$$

Performance metrics

Area Under Curve (AUC)

- the **ROC curve** is showing the performance of a classification model by plotting TPR and FPR
- the two-dimensional area underneath the ROC curve from (0,0) to (1,1) is called **Area Under Curve (AUC)**
- True Positive Rate (TPR) or **Recall** / **Sensitivity**:

$$TPR = \frac{TP}{TP + FN}$$

- False Positive Rate (FPR):

$$FPR = \frac{FP}{FP + TN}$$

Results of the paper vs. reproduced results
Geometric Mean

data set	paper	reproduced results
Moving Gaussian	0.7565	0.7956
SEA	0.9256	0.9388
Hyper Plane	0.5889	0.5751
Checkerboard	0.8123	0.6500
Electricity	0.7062	0.8026
Weather	0.6641	0.7150

Results of the paper vs. reproduced results

AUC

data set	paper	reproduced results
Moving Gaussian	0.8517	0.7964
SEA	0.9776	0.9385
Hyper Plane	0.7007	0.5747
Checkerboard	0.8876	0.6497
Electricity	0.8271	0.7964
Weather	0.7725	0.7162

Performance of DWMIL with F1-score

data set	f1-score
Moving Gaussian	0.8354
SEA	0.9398
Hyper Plane	0.5822
Checkerboard	0.6534
Electricity	0.7825
Weather	0.6447

Performance of DWMIL on new data sets

Forest Covertype

- DWMIL performs very good on this real-world data set

metric	value
gm	0.9222
f1	0.8040
auc	0.9211
rec	0.9129

Learn^{++}.NIE
Competitor

- Learn^{++} for Non-stationary and Imbalanced Environments
- modified algorithm of Learn^{++}.CDS
  - employs a different penalty constraint that forces the algorithm to balance predictive accuracy on all classes
  - uses a bagging based sub-ensemble for the minority class oversampling

For more details

[Gregory Ditzler and Robi Polikar] Incremental Learning of Concept Drift from Streaming Imbalanced Data. In IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pages 2283 – 2301. 10.1109/TKDE.2012.136, 2013.

DWMIL vs. Learn^{++}.NIE
Forest Covertype

- DWMIL performs better at this real-world data set

metric	DWMIL	Learn^{++}.NIE
gm	0.9222	0.8984
f1	0.8040	0.7838
auc	0.9211	0.8928
rec	0.9129	0.8616

DWMIL vs. Learn^{++}.NIE

Moving Gaussian

- Learn^{++}.NIE performs better at this data set

metric	DWMIL	Learn^{++}.NIE
gm	0.7956	0.9520
f1	0.8354	0.9568
auc	0.7964	0.9511
rec	0.7823	0.9724

DWMIL vs. Learn^{++}.NIE

SEA

- both methods perform equally good

metric	DWMIL	Learn^{++}.NIE
gm	0.9388	0.9729
f1	0.9398	0.9734
auc	0.9385	0.9727
rec	0.9460	0.9796

DWMIL vs. Learn^{++}.NIE

Hyper Plane

- Learn^{++}.NIE performs better at this data set

metric	DWMIL	Learn^{++}.NIE
gm	0.5751	0.9578
f1	0.5822	0.9594
auc	0.5747	0.9575
rec	0.5929	0.9662

DWMIL vs. Learn^{++}.NIE
Checkerboard

- Learn^{++}.NIE performs better at this data set

metric	DWMIL	Learn^{++}.NIE
gm	0.6500	0.9484
f1	0.6534	0.9496
auc	0.6497	0.9484
rec	0.6574	0.9490

DWMIL vs. Learn^{++}.NIE
Electricity

- Learn^{++}.NIE performs better at this data set

metric	DWMIL	Learn^{++}.NIE
gm	0.8026	0.9119
f1	0.7825	0.9064
auc	0.7964	0.9068
rec	0.7129	0.8607

DWMIL vs. Learn^{++}.NIE
Weather

- both methods perform equally good

metric	DWMIL	Learn^{++}.NIE
gm	0.7150	0.7731
f1	0.6447	0.7400
auc	0.7162	0.7662
rec	0.7234	0.6126

Conclusion

Authors of the paper

- concept drift and class imbalance are inevitable problems of learning from data streams
- DWMIL was proposed to solve these two problems
- the conducted experiments have shown that DWMIL
  - performs better compared with its counterparts
  - performs more efficiently compared with its counterparts

Conclusion
Reproduction

- our conducted experiments have shown that DWMIL
  - performs not always better compared with its counterparts
  - performs not always more efficiently compared with its counterparts
- Learn^{++}.NIE performed better compared to DWMIL most of the times
- Learn^{++}.NIE performed more efficiently compared to DWMIL most of the times

Thanks for your attention!

References

- [Yang Lu, Yiu-ming Cheung and Yuan Yan Tang] Dynamic Weighted Majority for Incremental Learning of Imbalanced Data Streams with Concept Drift. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pages 2393 – 2399. IJCAI-17, 2017.
- [Gregory Ditzler and Robi Polikar] Incremental Learning of Concept Drift from Streaming Imbalanced Data. In IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pages 2283 – 2301. 10.1109/TKDE.2012.136, 2013.