Online
Ensemble Approaches for Data Stream Mining

Leandro L. Minku
University of Birmingham (UK)
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• What to do if we actually have "little data"?
• How to handle class imbalance?
• Future directions
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• What to do if we actually have "little data"?
• How to handle class imbalance?
• Future directions
Data Streams

- Organisations have been gathering large amounts of data.
- The amount of data frequently grows over time [data streams].
Data Streams

We may wish to perform predictions based on such data streams.

Examples of data stream learning applications:
- credit card approval, spam filtering, electricity price prediction, etc.
Traditional Machine Learning Algorithms

Most machine learning algorithms operate in offline mode.

1. Training Phase
2. Testing Phase
Problems of Offline Learning

Problem:
Offline algorithms cannot process data streams, specially if amount of incoming data is high!
Problems of Offline Learning

Data streams usually suffer changes over time:

- **credit card approval**: customers may change their behaviour based on, e.g., world economic situation;
- **spam detection**: new spam strategies may be developed;
- **electricity price prediction**: world climate changes.

**Problem:**
Offline algorithms cannot incorporate strategies to deal with changes!
Online Learning Algorithms

- Training and testing phase co-occur.

1. Training Phase
2. Testing Phase

- Process each training example separately upon arrival and then discard it.
  - Can update models with new data.
  - Require less memory and use less running time.
  - Can incorporate strategies to deal with changes.
Good Approach for Online Learning in Changing Environments

An ideal online learning algorithm for changing environments should:

- maximise accuracy when the environment is stable;
- minimise the drop in accuracy when there are changes;
- quickly recover from changes.

It should work for different types of change!
Online Ensembles of Learning Machines

- Ensembles: sets of learning machines grouped together.
- Aim: to improve predictive performance.
- Key: diversity, i.e., models give different errors on the same examples.

$$\text{ensemble estimation} = \sum_{i} \text{estimation}_i$$
[Video -- BBC The Code -- Wisdom of the Crowd]
https://youtu.be/iOucwX7Z1HU

[A test on history and politics]
Ensemble Diversity

Ensemble diversity is also helpful for dealing with changes.
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• How to handle class imbalance?
• What to do if we actually have "little data"?
• Future directions
Concept Drifts

Changes in the underlying distribution of the problem $p(x,w)$:

- Unconditional probability density function (pdf) $p(x)$
- Posterior probabilities $P(w_i|x)$, $i = 1,2,...,c$

or

- Prior probabilities of the classes $P(w_1),...,P(w_c)$, where $c$ is the number of classes
- Class-conditional pdfs $p(x|w_i)$, $i = 1,...,c$

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Why Adopting this Definition?

Some authors consider concept drift as a change in the posterior probabilities $P(w_i|x), i = 1, 2, ..., c$

- Change in the true class boundaries.

However, even if changes solely in $p(x)$ can cause a need for change in the learnt class boundaries!
Characterising Concept Drifts in Isolation

Class severity: amount of changes in the target class of examples.

Possible measure: percentage of the input space which has its target class changed after the drift is complete.

Example: housing pricing reading preferences.

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Feature severity: amount of changes in the distribution of the input attributes.

Possible measure: percentage of the input space that has its unconditional pdf modified.

Example: change in the number of articles with text about elections.

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Speed: inverse of the drifting time (time for the drift to complete).

Possible drifting time measure: the number of time steps taken for a new concept to completely replace the old one.

Example: machine starting to malfunction.

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Characterising Sequences of Concept Drifts

Frequency: how often concept drifts happen.

Possible measure: number of time steps between the start of two consecutive drifts.

Example: change in electricity demand every season.

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Characterising Sequences of Concept Drifts

**Recurrence**: whether the concept can change to previous (or similar to previous) concepts.

**Example**: change to a previous electricity demand.

There is a notion of severity here too.

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Predictability: whether the sequence of changes can be learnt to predict future changes.

Example: people buying more and more selfie sticks.

L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• How to handle class imbalance?
• What to do if we actually have "little data"?
• Future directions
Data Sets

• Artificial vs real world data:
  • [http://www.cs.bham.ac.uk/~minkull/opensource/ArtificialConceptDriftDataSets.zip](http://www.cs.bham.ac.uk/~minkull/opensource/ArtificialConceptDriftDataSets.zip)
  • Test on different types of concept drift.
Computing Performance

• Performance on a separate test set.

Time
Computing Performance

- Performance on the next $X$ examples.
Computing Performance

• Prequential performance, e.g., accuracy:

\[ acc(t) = \begin{cases} 
acc_{ex}^{(t)}, & \text{if } t = 1 \\
\frac{(t-1) \cdot acc(t-1) + acc_{ex}^{(t)}}{t}, & \text{otherwise}
\end{cases} \]

where \( acc_{ex}^{(t)} \) is the error (0 or 1) on the current training example \( ex \) before its learning.
Computing Performance

- Prequential performance, e.g., accuracy:
  \[
  acc(t) = \begin{cases} 
  acc_{ex}^{(t)}, & \text{if } t = 1 \\
  \frac{(t-1) \cdot acc^{(t-1)} + acc^{(t)}}{t}, & \text{otherwise}
  \end{cases}
  \]
  where \( acc_{ex}^{(t)} \) is the error (0 or 1) on the current training example \( ex \) before its learning.

- Prequential performance with forgetting mechanism.
- Reset prequential performance upon changes.
- ...

Are the Performances of Different Approaches Really Different?

Specially when our predictors are stochastic...

Statistical tests with Holm-Bonferroni corrections at certain points of the learning.

- Wilcoxon tests (sign-rank or rank-sum)
- Matlab: \[p, h\] = signrank(a, b)
- Matlab: \[p, h\] = ranksum(a, b)
JELLY BEANS CAUSE ACNE!

SCIENTISTS! INVESTIGATE!

BUT WE'RE PLAYING MINECRAFT!
...FINE.

WE FOUND NO LINK BETWEEN JELLY BEANS AND ACNE (P > 0.05).

THAT SETTLES THAT.
I HEAR IT'S ONLY A CERTAIN COLOR THAT CAUSES IT.

SCIENTISTS!

WE FOUND NO LINK BETWEEN GREY JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLUE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TEAL JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BROWN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN ORANGE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PINK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN MAGENTA JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN CYAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN YELLOW JELLY BEANS AND ACNE (P > 0.05).
WE FOUND A LINK BETWEEN ORANGE JELLY BEANS AND ACNE (P < 0.05).

GREEN JELLY BEANS LINKED TO ACNE!
95% CONFIDENCE

ONLY 5% CHANCE OF COINCIDENCE!

Source: http://xkcd.com/882/
Are the Performances of Different Approaches Really Different?

Specially when our predictors are stochastic...

Confidence intervals throughout the learning.

- Given a set of estimations for the same example, what interval would contain estimation values that would not be unusual?
Are the Performances of Different Approaches Really Different?

Specially when our predictors are stochastic...

Standard deviations throughout the learning.
Are the Performances of Different Approaches Really Different?

Statistical tests for overall performance across data sets.

• Friedman tests.


ANOVA for analysis of impact of factors on performance.
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• What to do if we actually have "little data"?
• How to handle class imbalance?
• Future directions
Ensemble Diversity

• Can help to improve predictive performance in static environments.
• Can it also help to deal with changes? When and how?

... we need an algorithm that allows us to tune the level of diversity.
Online Bagging

Online bagging creates diverse models by training them with different data samples.

### Algorithm

**Input:** an ensemble with $M$ base learners, and current training example $(x_t, y_t)$.

```
for each base learner $f_m$ ($m = 1, 2, \ldots, M$) do
    set $K \sim \text{Poisson}$
    update $f_m$ $K$ times
end for
```

Changing sampling rate will lead to different levels of diversity.

---

Experimental Study -- Impact of Lambda on Diversity

Objective: to understand the impact of lambda on diversity

Base learners: 25 DTs

Diversity measure: Q statistic (range [-1, 1])

\[ Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \]

where \( N^{a,b} \) is the number of training examples for which the classification given by \( D_i \) is \( a \) and the classification given by \( D_k \) is \( b \), 1 represents a correct classification and 0 represents a misclassification.

Higher values = less diversity

Data Sets

Experiments on 6 x 3 x 3 artificial data sets.

<table>
<thead>
<tr>
<th>Probl.</th>
<th>Equation</th>
<th>Fixed Values</th>
<th>Before $\rightarrow$ After Drift</th>
<th>Sev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle</td>
<td>$(x-a)^2 + \frac{(y-b)^2}{r^2} \leq$</td>
<td>$a = 0.5$, $b = 0.5$</td>
<td>$r = 0.2 \rightarrow 0.3$, $r = 0.2 \rightarrow 0.4$, $r = 0.2 \rightarrow 0.5$</td>
<td>16%, 38%, 66%</td>
</tr>
<tr>
<td>SineV</td>
<td>$y \leq \frac{a \sin(bx + c) + d}{c}$</td>
<td>$a = 1$, $b = 1$, $c = 0$</td>
<td>$d = -2 \rightarrow 1$, $d = -5 \rightarrow 4$, $d = -8 \rightarrow 7$</td>
<td>15%, 45%, 75%</td>
</tr>
<tr>
<td>SineH</td>
<td>$y \leq \frac{a \sin(bx + c) + d}{c}$</td>
<td>$a = 5$, $d = 5$, $b = 1$</td>
<td>$c = 0 \rightarrow -\pi/4$, $c = 0 \rightarrow -\pi/2$, $c = 0 \rightarrow -\pi$</td>
<td>36%, 57%, 80%</td>
</tr>
<tr>
<td>Line</td>
<td>$y \leq -a_0 + a_1 x_1$</td>
<td>$a_1 = 0.1$</td>
<td>$a_0 = -0.4 \rightarrow -0.55$, $a_0 = -0.25 \rightarrow -0.7$, $a_0 = -0.1 \rightarrow -0.8$</td>
<td>15%, 45%, 70%</td>
</tr>
<tr>
<td>Plane</td>
<td>$y \leq -a_0 + a_1 x_1 + a_2 x_2$</td>
<td>$a_1 = 0.1$, $a_2 = 0.1$</td>
<td>$a_0 = -2 \rightarrow -2.7$, $a_0 = -1 \rightarrow -3.2$, $a_0 = -0.7 \rightarrow -4.4$</td>
<td>14%, 44%, 74%</td>
</tr>
<tr>
<td>Bool</td>
<td>$(color eq_{1a} \ op_1 \ shape eq_{2 b}) \ op_2 \ size eq_{3 c}$</td>
<td>$eq_{1,2,3}$</td>
<td>$a = R, op_1 \land \rightarrow \lor$, $a = R, b = R$, $op_1 \land \rightarrow \lor$</td>
<td>11%, 44%, 67%</td>
</tr>
</tbody>
</table>

Three different drifting times: 1 time step, $0.25N$, 0.5N

Total no. of examples: 2N.

N=1000 for all but boolean, where it was 500.
Impact of Lambda on Diversity

- ANOVA to check impact of lambda, type of drift and time on Q statistics.
- **Lambda has a strong effect size on Q Statistics** (eta-squared always > 0.90 and usually > 0.97).

Main effect of Lambda on Q Statistics for Circle Data
Experimental Study -- Impact of Diversity on Accuracy

Objective: to understand the impact of diversity (lambda) on the ability to (1) perform well in static periods, (2) minimise drop in accuracy and (3) recover from concept drifts.

Comparisons: high vs low diversity online bagging ensembles.

Base learners: 25 DTs.

Performance measure: classification error on separate data set.

Data sets: same as in preliminary study.
Impact of Diversity on Ability to Deal with Changes

- **ANOVA** to check impact of diversity (lambda), and other factors (type of drift and time step) on accuracy.
- Diversity and its interaction with time step had large effect size on accuracy.

Different levels of diversity can help to deal with different environment conditions:

- **Low diversity** is good for stable periods.
- **High diversity** reduces the impact of changes.

* Sample result for Circle problem.
Experimental Study -- Recovering from Changes

Objective: to determine what strategy to use for recovering from changes.

Comparisons:
- Keep old low diversity ensemble.
- Keep old high diversity ensemble, but enforce low diversity in it.
- Create new low diversity ensemble.
- Create new high diversity ensemble.

Base learners: 25 DTs

Performance measure: prequential accuracy reset upon drift.
Recovering from Changes

Different types of change require different strategies for recovery:

- **Very severe change**: create new low diversity ensemble from scratch (similarly to strategy done in the literature).

* Sample result for Circle Problem
Recovering from Changes

Different types of change require different strategies for recovery:

- Low severity change: old high diversity ensemble learning new situation with low diversity.
- It can use information previously learnt to aid learning the new situation!

* Sample result for Circle Problem
Recovering from Changes

Different types of change require different strategies for recovery:

- **Slow changes**: keep old ensembles until the change is completed.

* Sample result for Circle Problem (low severity on left, high severity on right)
Outline

• Introduction
• What types of change exist?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• How to handle class imbalance?
• What to do if we actually have "little data"?
• Future directions
Diversity for Dealing with Drifts (DDD)

Before change detection

Early Drift Detection Method


\[
(p'_i + 2 \cdot s'_i)/(p'_\text{max} + 2 \cdot s'_{\text{max}}) < \beta
\]

Change Detection Method

X X X X X X X

Time

p' = avg distance between errors

s' = standard deviation

Predictions


www.cs.bham.ac.uk/~minkull

Ensemble Approaches for Data Stream Mining

47
Diversity for Dealing with Drifts (DDD)

Before change detection

Low diversity ensemble:
- Learning
- Predictions

High diversity ensemble:
- Learning

After change detection

Old low diversity ensemble:
- Learning
- Predictions

Old High diversity ensemble:
- Learning with low diversity
- Predictions

New low diversity ensemble:
- Learning
- Predictions

New High diversity ensemble:
- Learning

Experimental Study

Objectives:

• Validate DDD, checking if it has considerably good accuracy in the presence and absence of drifts.
• Identify the types of drift to which it works better.
• Explain its behaviour.

Comparisons:

• DDD vs Early Drift Detection Method (EDDM).

• DDD vs Dynamic Weighted Majority (DWM).

• DDD vs Online bagging.

Base learners:

• 25 DTs for artificial data.
• 25 MLPs or 25 NB for real world data.
Data Sets

Experiments on 6 x 3 x 3 artificial and 3 real world data sets.

<table>
<thead>
<tr>
<th>Probl.</th>
<th>Equation</th>
<th>Fixed Values</th>
<th>Before→After Drift</th>
<th>Sev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle</td>
<td>$(x - a)^2 + \frac{(y - b)^2}{r^2} \leq$</td>
<td>$a = .5 \quad b = .5$</td>
<td>$r = .2 \rightarrow .3$, $r = .2 \rightarrow .4$, $r = .2 \rightarrow .5$</td>
<td>16%, 38%, 66%</td>
</tr>
<tr>
<td>SineV</td>
<td>$y \leq \frac{a \sin(bx + c) + d}{d}$</td>
<td>$a = 1 \quad b = 1 \quad c = 0$</td>
<td>$d = -2 \rightarrow 1$, $d = -5 \rightarrow 4$, $d = -8 \rightarrow 7$</td>
<td>15%, 45%, 75%</td>
</tr>
<tr>
<td>SineH</td>
<td>$y \leq \frac{a \sin(bx + c) + d}{d}$</td>
<td>$a = 5 \quad d = 5 \quad b = 1$</td>
<td>$c = 0 \rightarrow -\pi/4$, $c = 0 \rightarrow -\pi/2$, $c = 0 \rightarrow -\pi$</td>
<td>36%, 57%, 80%</td>
</tr>
<tr>
<td>Line</td>
<td>$y \leq -a_0 + \frac{a_1 x_1}{d}$</td>
<td>$a_1 = .1$</td>
<td>$a_0 = -.4 \rightarrow -.55$, $a_0 = -.25 \rightarrow -.7$, $a_0 = -.1 \rightarrow -.8$</td>
<td>15%, 45%, 70%</td>
</tr>
<tr>
<td>Plane</td>
<td>$y \leq -a_0 + \frac{a_1 x_1 + a_2 x_2}{d}$</td>
<td>$a_1 = .1 \quad a_2 = .1$</td>
<td>$a_0 = -2 \rightarrow -2.7$, $a_0 = -1 \rightarrow -3.2$, $a_0 = -.7 \rightarrow -4.4$</td>
<td>14%, 44%, 74%</td>
</tr>
<tr>
<td>Bool</td>
<td>$(color eq_{1a} \quad op_1 \quad shape eq_{2 b} \quad op_2 \quad size eq_{3 c})$</td>
<td>$c = S \lor M \lor L$</td>
<td>$a = R$, $b = R \lor T$, $op_1 \land \rightarrow \lor$</td>
<td>11%, 44%, 67%</td>
</tr>
</tbody>
</table>

Three different drifting times:
1 time step
0.25N
0.5N

Total no. of examples: 2N.

N=1000 for all but boolean, where it was 500.
Data Sets

Network intrusion detection (KDD Cup 1999):

- 494,020 examples
- 41 input attributes (length of the connection, the type of protocol, the network service on the destination, etc.)
- 1 target class: attack or a normal connection.

Electricity price prediction:

- 45,312 examples
- 4 input attributes (time stamp, day of the week and 2 electricity demand values)
- 1 target class: increase / decrease in price.

Credit card approval:

- 50,000 examples corresponding to a 1 year period.
- 27 input attributes (sex, age, marital status, profession, income, etc.)
- 1 target class: “good” or “bad” client.
Validating DDD and Explaining its Behaviour

- Low speed and false alarms: successful use of old low diversity ensemble.
- High severity: successful selection of new low diversity ensemble.
Validating DDD on Real World Data

![Graph showing Accuracy + Std Deviation over time steps for Electricity (NB) and Credit Card (MLP).]
Type of Drifts for Which DDD Behaves Better

T-tests with Holm-Bonferroni corrections after drifts:

<table>
<thead>
<tr>
<th></th>
<th>Win</th>
<th>Tie</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDD vs EDDM</td>
<td>45%</td>
<td>48%</td>
<td>7%</td>
</tr>
<tr>
<td>DDD vs DWM</td>
<td>59%</td>
<td>25%</td>
<td>15%</td>
</tr>
</tbody>
</table>

DDD performs better specially for:

- Low speed drifts.
- Low severity drifts.

DDD performed worse when:

- Drift was very fast and severe + inaccurate initial weights.
- Drift had almost no effect.
Summary of DDD's Results

• DDD presents good accuracy both in the presence and absence of drifts. DDD rarely presents worse accuracy than other approaches.

• DDD successfully uses weights to choose good ensembles for the predictions.

• DDD is particularly good for low severity or low speed drifts.

DDD is successful in achieving robustness to different types of drifts.
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• What to do if we actually have "little data"?
• How to handle class imbalance?
• Future directions
Online Ensemble Learning for Software Effort Estimation (SEE)

Some applications have little data from within the organisation that we are interested in.

Estimation of the effort required to develop a software project.

• Effort is measured in person-hours, person-months, etc.
• Based on features such as required reliability, programming language, development type, team expertise, etc.
• Main factor influencing project cost.
• Overestimation vs underestimation.
Example of Underestimation

Nasa cancelled its incomplete Check-out Launch Control Software project after the initial $200M estimate was exceeded by another $200M.

Machine Learning for SEE

Predictive models can be created based on completed software projects.

Challenges:

- Companies have changing environments.
- Little *within-company* data.
- Cross-company data are available.
Experimental Study

Objectives:

• Check whether cross-company data can be beneficial for software effort estimation.

Comparisons:

• Cross-company vs within-company models.

Performance measure:

• Mean absolute error over next 10 examples.

Learners:

• Regression trees.
Data Sets

- **ISBSG’2000**: 119 WC, 168 CC.
- **ISBSG’2001**: 69 WC, 224 CC.
- **ISBSG**: 187 WC, 826 CC.

- 3 CC subsets (pre-existing data) based on distribution of productivity rate.
- **Input attributes**: development type, language type, development platform, functional size.
- **Output**: software effort in person-hours

- K-NN imputation for missing attributes.
Data Sets

- **CocNasaCoc81**: 60 WC Nasa projects 1980s–1990s. 63 CC projects up to 1980.
- **CocNasaCoc81Nasa93**: additional 93 Nasa projects 1970s–1980s, considered CC.

- **Input attributes**: cocomo cost drivers and LOC.
- **Output**: software effort in person-months.

- 3 CC subsets for CocNasaCoc81, 2 CC subsets for CocNasaCoc81Nasa93.
Analysis

• If a CC-RT obtains better performance than the WC-RT at a certain time step, it is potentially beneficial at this time step. Otherwise, it is detrimental.

• Different companies represent different concepts.

• Changes can make a company behave more or less similar to another.

How to make use of that to improve performance?
Dynamic Cross-Company Learning (DCL)

Cross-company (CC) $m$ training sets with different productivity (completed projects)

Within-company (WC) incoming training data (completed projects arriving with time)

CC Model 1

CC Model 2

…

CC Model M

WC Model

DCL learns a weight to reflect the suitability of CC models.

For each new training project,
If model is not a winner, multiply its weight by $\beta$ ($0 < \beta < 1$)

Experimental Study

Objectives:
• Check whether DCL can improve performance over within-company models.

Comparisons:
• DCL vs within-company RT (no drift handling).
• DCL vs Dynamic Weighted Majority using WC data (WC-DWM).
• DCL vs Dynamic Weighted Majority using WC+CC data (CC-DWM).

Performance measure:
• Mean absolute error (over next 10 examples) = \( \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| / n \)
• Standardised accuracy = (1 - MAE / MAEguess) * 100
• Effect size = |MAEc - MAE| / MAEc

Base learners:
• Regression trees.

Data sets:
• Same as in previous study.
Overall Performance

- Friedman (MAE across data sets): overall MAE of the approaches is statistically significantly different (p-value = 0.0001).
- Post-hoc against RT: DCL is better than RTs; no evidence to reject null hypothesis for others.
- Effect size (RT as the control approach): increases as we go from WC-DWM (low), to CC-DWM (low-medium) and DCL (low-high).

Dealing with drifts decreases overall MAE slightly.
Using CC decreases it further.
With DCL the difference becomes statistically significant.
Performance Throughout Time

- Considerable improvement over periods of several months. Improvements in general coincide with the periods when some CC model is beneficial.
- Even though DCL rarely performs worse, there is still some room for improvement.

DCL uses CC models only when they are directly useful.

ISBSG'2001 (≈ 1.5 year)
Dynamic Cross-Company Mapped Model Learning (Dycom)

Dycom learns functions to map CC models to the context of our single company.

Mapping Training Examples

Mapping one CC SEE model $\hat{f}_A(x) = \hat{g}_{BA}(\hat{f}_B(x))$:

- **$C_B$ (CC) Training Data**
  - Training Examples
  - Learning $\hat{f}_B$
  - Function Learnt
  - Estimation Enquiries
  - Effort Estimations $\hat{f}_B(x)$

- **$C_A$ (WC) Training Data**
  - Training Examples $(x, y)$
  - Create Mapping Training Examples
  - Training Examples $(\hat{f}_B(x), y)$
  - Mapping Training Data

- **$\hat{g}_{BA}$**
  - Function Learnt
  - Learning $\hat{g}_{BA}$
  - Training Examples
Dycom's Mapping Function

\[ \hat{f}_A(x) = \hat{g}_{BiA}(\hat{f}_{Bi}(x)) = \hat{f}_{Bi}(x) \cdot b_i \]

\[ b_i = \begin{cases} 
1, & \text{if no mapping training example has been received yet;} \\
\frac{y}{\hat{f}_{Bi}(x)}, & \text{if } (\hat{f}_{Bi}(x), y) \text{ is the first mapping training example;} \\
lr \cdot \frac{y}{\hat{f}_{Bi}(x)} + (1 - lr) \cdot b_i, & \text{otherwise.}
\end{cases} \]

where \( lr \) is a smoothing factor that allows tuning the emphasis on more recent examples.
Objective:
• To determine whether Dycom is able to maintain or improve performance in comparison to a corresponding WC model while using less WC training examples than this model.

Comparisons:
• RT vs Dycom-RT.

Base learners:
• RT

Dycom is set to use $p = 10$, i.e., it is trained with only 10% of the WC training examples used by RT, and $lr = 0.1$
Data Sets

**ISBSG’2000:** 119 WC, 168 CC.
**ISBSG’2001:** 69 WC, 224 CC.
**ISBSG:** 187 WC, 826 CC.
  - Input attributes: development type, language type, development platform and functional size.
  - Target: effort in person-hours.

**CocNasaCoc81:** 60 WC Nasa projects, 63 CC projects. Input attributes: 15 cost drivers and KLOC.
  - Target: effort in person-months.

  - Target: effort in person-hours.

CC projects were divided into 3 subsets according to their productivity.
Performance Measures

\[ MAE = \frac{1}{T} \sum_{i=1}^{T} |\tilde{y}_i - y_i|; \]

\[ \text{StdDev} = \text{standard deviation of MAE across time steps.} \]

\[ SA = \left(1 - \frac{MAE}{MAE_{\text{guess}}}\right) \cdot 100, \]

\[ RMSE = \sqrt{\frac{\sum_{i=1}^{T} (\tilde{y}_i - y_i)^2}{T}}; \]

\[ Corr = \frac{\sum_{i=1}^{T} (\tilde{y}_i - \tilde{y})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{T} (\tilde{y}_i - \tilde{y})^2 \sqrt{\sum_{i=1}^{T} (y_i - \bar{y})^2}}, \]

where \( \tilde{y} \) and \( \bar{y} \) are the average predicted and average actual efforts, respectively;

\[ LSD = \sqrt{\frac{\sum_{i=1}^{T} \left(e_i + \frac{s_i^2}{2}\right)^2}{T-1}}, \]

where \( s^2 \) is an estimator of the variance of the residual \( e_i \) and \( e_i = \ln y_i - \ln \hat{y}_i; \)

Calculated over next 10 examples.
## Overall Performance

Dycom’s MAE (and SA), StdDev, RMSE, Corr and LSD were always similar or better than RT’s (Wilcoxon tests with Holm-Bonferroni corrections).

<table>
<thead>
<tr>
<th>Database</th>
<th>Approach</th>
<th>MAE</th>
<th>StdDev</th>
<th>SA</th>
<th>RMSE</th>
<th>Corr</th>
<th>LSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>KitchenMax</td>
<td>RT</td>
<td>2441.0241</td>
<td>2838.2375</td>
<td>30.1782</td>
<td>4850.3387</td>
<td>0.4350</td>
<td>1.2221</td>
</tr>
<tr>
<td></td>
<td>Dycom-RT</td>
<td>2208.6522</td>
<td>2665.4276</td>
<td>36.8249</td>
<td>4287.4476</td>
<td>0.6416</td>
<td>0.8809</td>
</tr>
<tr>
<td></td>
<td>P-value</td>
<td>3.82E-11</td>
<td>6.35E-01</td>
<td>1.46E-12</td>
<td>1.62E-16</td>
<td>4.25E-21</td>
<td></td>
</tr>
<tr>
<td>CocNasaCoc81</td>
<td>RT</td>
<td>319.4572</td>
<td>250.2325</td>
<td>33.1366</td>
<td>477.2357</td>
<td>0.6427</td>
<td>0.8623</td>
</tr>
<tr>
<td></td>
<td>Dycom-RT</td>
<td>161.7917</td>
<td>105.7591</td>
<td>66.1365</td>
<td>243.6504</td>
<td>0.8885</td>
<td>0.6671</td>
</tr>
<tr>
<td></td>
<td>P-value</td>
<td>4.04E-06</td>
<td>1.40E-11</td>
<td>5.95E-08</td>
<td>4.12E-07</td>
<td>8.82E-04</td>
<td></td>
</tr>
<tr>
<td>ISBSG2000</td>
<td>RT</td>
<td>2753.3726</td>
<td>1257.4586</td>
<td>37.0471</td>
<td>4133.1006</td>
<td>0.3554</td>
<td>1.4592</td>
</tr>
<tr>
<td></td>
<td>Dycom-RT</td>
<td>2494.6639</td>
<td>1249.8400</td>
<td>42.9622</td>
<td>3741.8009</td>
<td>0.4515</td>
<td>1.1589</td>
</tr>
<tr>
<td></td>
<td>P-value</td>
<td>4.72E-02</td>
<td>1.01E-01</td>
<td>1.83E-01</td>
<td>8.73E-02</td>
<td>1.27E-06</td>
<td></td>
</tr>
<tr>
<td>ISBSG2001</td>
<td>RT</td>
<td>3621.9598</td>
<td>1367.9603</td>
<td>11.9270</td>
<td>5149.6267</td>
<td>0.1658</td>
<td>1.8110</td>
</tr>
<tr>
<td></td>
<td>Dycom-RT</td>
<td>2543.9495</td>
<td>1165.8591</td>
<td>38.1403</td>
<td>3581.6573</td>
<td>0.5691</td>
<td>1.2447</td>
</tr>
<tr>
<td></td>
<td>P-value</td>
<td>3.21E-06</td>
<td>4.16E-01</td>
<td>7.88E-06</td>
<td>2.29E-10</td>
<td>6.24E-08</td>
<td></td>
</tr>
<tr>
<td>ISBSG</td>
<td>RT</td>
<td>3253.9349</td>
<td>2476.0512</td>
<td>46.2891</td>
<td>4872.9193</td>
<td>0.4412</td>
<td>1.3475</td>
</tr>
<tr>
<td></td>
<td>Dycom-RT</td>
<td>3122.6603</td>
<td>2227.9812</td>
<td>48.4560</td>
<td>4473.6527</td>
<td>0.5817</td>
<td>1.0378</td>
</tr>
<tr>
<td></td>
<td>P-value</td>
<td>5.56E-02</td>
<td>3.54E-01</td>
<td>4.18E-02</td>
<td>1.90E-09</td>
<td>2.99E-12</td>
<td></td>
</tr>
</tbody>
</table>
Dycom can achieve similar / better performance while using only 10% of WC data.
Summary of Dycom's Results

Dycom is able to maintain or improve performance in comparison to a corresponding WC model while using less WC training examples than this model.
Insights Provided by Dycom

• Mapping functions learnt by Dycom explain the relationship between the effort of different companies for the same projects.

• The factor $b_i$ can be plotted to visualise that.

• It can show the need for strategic decision making towards improvement of productivity.

• It can be used to monitor the success of strategies being adopted.
Dycom Insights on Productivity

\[ \hat{f}_A(x) = \hat{f}_{B_i}(x) \cdot b_i \]

- Initially, our company needs initially \(2\)x effort than company red.
- Later, it needs only \(1.2\)x effort.
Dycom Insights on Productivity

\[ \hat{f}_A(x) = \hat{f}_{B_i}(x) \cdot b_i \]

- Our company needs 2x effort than company red.
- How to improve our company?
Analysing Project Data

Number of projects with each feature value for the 20 CC projects from the medium productivity CC section and the first 20 WC projects:

<table>
<thead>
<tr>
<th>Feature / Value</th>
<th>Lang. exp</th>
<th>Virtual mach. exp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>WC</td>
</tr>
<tr>
<td>Very low</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Nominal</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>High</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Very high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Extremely high</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Both the company and the medium CC section frequently use employees with high programming language experience.
Analysing Project Data

Number of projects with each feature value for the 20 CC projects from the medium productivity CC section and the first 20 WC projects:

<table>
<thead>
<tr>
<th>Feature / Value</th>
<th>Lang. exp</th>
<th>Virtual mach. exp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>WC</td>
</tr>
<tr>
<td>Very low</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Nominal</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>High</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Very high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Extremely high</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Medium CC section uses more employees with high virtual machine experience. So, this is more likely to be a problem for the company. Sensitivity analysis and project manager knowledge could help to confirm that.
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• What to do if we actually have "little data"?
• How to handle class imbalance?
• Future directions
Online Class Imbalance Learning

Online learning: algorithms that process each incoming training example separately and then discard it.

Class imbalance learning: algorithms to learn data with skewed class distributions.

Examples of applications:

- Credit card approval, intrusion detection in computer networks, fault diagnosis, spam detection, etc.
Challenges

Difficulty of **class imbalance learning:**
- Minority class cannot draw equal attention to the majority class.
- Algorithms struggle to perform well on the minority class.

[weka example]

In **online class imbalance learning:**
- Whole picture is not available for evaluating class imbalance status.
- Imbalance status may change over time. E.g., news articles on elections.
Concept Drifts

Changes in the underlying distribution of the problem $p(x, w)$:

- Unconditional probability density function (pdf) $p(x)$
- Posterior probabilities $P(w_i | x)$, $i = 1, 2, \ldots, c$

or

- Prior probabilities of the classes $P(w_1), \ldots, P(w_c)$, where $c$ is the number of classes
- Class-conditional pdfs $p(x | w_i)$, $i = 1, \ldots, c$
Class Imbalance Vs Class-Conditional Changes

Class imbalance and class-conditional changes have to be treated differently.

E.g., resetting the system when a class changes from majority to minority could be detrimental!
Online Class Imbalance Learning Framework

Refer to class-conditional changes as concept drift for simplicity.

The framework allows treating class imbalance and concept drift differently.
Online Class Imbalance Learning Framework
Defining Class Size Online

• Suppose an incoming sequence of examples \((x_t, c_t)\).

• Time-decayed class size:

\[
\begin{align*}
    w_k^{(t)} &= \theta w_k^{(t-1)} + (1 - \theta) \left[ (x_t, c_k) \right], \quad (k = 1, \ldots, N)
\end{align*}
\]

where \([(x_t, c_k)] = 1\) if true class label of \(x_t\) is \(c_k\), and 0 otherwise; and \(\theta\) is a decay factor.
Monitoring Class Size -- No Change

\[ \theta = 0.9 \]
Monitoring Class Size -- Change With Low Severity
Monitoring Class Size -- Change With High Severity

θ = 0.9
Class Imbalance Detector

The data is imbalanced currently, if there are any two classes $c_i$ and $c_j$, satisfying:

- Class percentage difference $w_i - w_j > \delta_1$
- Recall difference $R_i - R_j > \delta_2$

Then, class $c_i$ is the majority; class $c_j$ is the minority.

Time-decayed recall (TP / P):

$$R_k^{(t)} = \eta' R_k^{(t-1)} + (1 - \eta') [x \leftarrow c_k]$$

where $[x \leftarrow c_k] = 1$ if $x$ belongs to $c_k$ and is correctly classified as $c_k$;

0 if $x$ belongs to $c_k$ but is incorrectly classified; and $\eta'$ is a decay factor.
Online Class Imbalance Learning Framework
Resampling-Based Ensemble Approaches

Oversampling Online Bagging (OOB) and Undersampling Online Bagging (UOB).

Online Bagging:

Input: an ensemble with $M$ base learners, and current training example $(x_t, y_t)$.

for each base learner $f_m$ ($m = 1, 2, \ldots, M$) do
    set $K \sim \text{Poisson}$
    update $f_m$ $K$ times
end for

OOB and UOB vary this parameter to change the sampling rate.
Oversampling the Minority Class

Oversample

www.cs.bham.ac.uk/~minkull
Ensemble Approaches for Data Stream Mining
Undersampling the Majority Class

Undersample
Resampling-Based Ensemble Approaches

Input: an ensemble with $M$ base learners, current training example $(x_t, y_t)$, and current class size $w(t) = (w_+, w_-)$.

for each base learner $f_m$ ($m = 1, 2, \ldots, M$) do
  if $y_t = +1$ and $w_+ < w(t)$ for OOB
    set $K \sim \text{Poisson} \left( \frac{w(t)}{w_+} \right)$
  else if $y_t = -1$ and $w_- > w(t)$ for OOB
    set $K \sim \text{Poisson} \left( \frac{w(t)}{w_-} \right)$
  else
    set $K \sim \text{Poisson} (1)$
  end if
  update $f_m$ $K$ times
end for

$+1$ is a "minority"
$-1$ is a "majority"
undersample ($\lambda < 1$)

$+1$ is a "majority"
$-1$ is a "minority"
undersample ($\lambda < 1$)

no resampling as $y_t$ is a minority

S. Wang, L. Minku and X. Yao. Resampling-Based Ensemble Methods for Online Class Imbalance Learning, IEEE Transactions on Knowledge and Data Engineering, 14p., 2014 (in press)
Advantages of the Resampling-Based Ensemble Approaches

• **Resampling-based:** independent of the underlying learning algorithm.
• **Time-decayed class size:** automatically estimates imbalance status and decides the resampling rate.
• **Ensemble-based:** combines multiple classifiers to improve performance.
Previous Work on Online Class Imbalance Learning

Little work on online class imbalance learning.

Resampling-based approaches:
• Assume that imbalance rate is known a priori.
• Assume fixed imbalance rate.

Cost-based approaches:
• Perceptron-based methods.
• Costs adjusted based on a window of examples or on a pre-existing validation set.

Only evaluated under static imbalance conditions!
Experimental Study

Objectives:

• Check whether OOB and UOB are able to automatically adjust to changing imbalance rates.

• Provide a thorough analysis of the impact of class imbalance in online learning, considering both static and dynamic class imbalance rates.
Analysis in Static Data Streams

Research questions:

1. To what extent does resampling in OOB and UOB help dealing with class imbalance online?
2. How do they perform in comparison with previous approaches?
**Experimental Setup**

**Data sets:**

- 12 two-class data streams with different data distributions and imbalance rates (1000 examples):
  - Each class follows Gaussian distribution, 2 inputs.
  - 4 imbalance rates: 5%, 10%, 20% and 30%.
  - 3 minority-class distributions: safe, borderline, rare/outlier.

Experimental Setup

Data sets:

- 12 two-class data streams with different data distributions and imbalance rates (1000 examples):
  - Each class follows Gaussian distribution, 2 inputs.
  - 4 imbalance rates: 5%, 10%, 20% and 30%.
  - 3 minority-class distributions: safe, borderline, rare/outlier.
- 2 real world problems (1000 examples, 10% imbalance rate):
  - Gearbox fault detection.
  - Sensor contaminant detection in smart buildings.

Performance measure:

- Prequential minority class recall and G-mean.
- Wilcoxon sign-rank tests with holm-bonferroni corrections for comparing final step performance.
Evaluating Performance in Class Imbalance Learning

Measures such as accuracy can be misleading:

Majority: 94%
Minority: 6%

Consider that a classifier correctly classifies all majority examples and incorrectly classifies all minority examples.

What would the accuracy of this classifier be? 94%
Evaluating Performance in Class Imbalance Learning

Recall on each class separately:

- \( \frac{TP}{P} \)

Majority: 94%
Minority: 6%

Consider that a classifier correctly classifies all majority examples and incorrectly classifies all minority examples.

What would the recall on each class be?

majority: \( \frac{94}{94} = 100\% \)
minority: \( \frac{0}{6} = 0\% \)
Evaluating Performance in Class Imbalance Learning

G-mean: \( \sqrt{\frac{TP}{P} \cdot \frac{TN}{N}} \)

Majority: 94%
Minority: 6%

Consider that a classifier correctly classifies all majority examples and incorrectly classifies all minority examples.

What would the g-mean be? \( 100\% \times 0\% = 0 \)

And what if half of the minority class were classified correctly?

\( \sqrt{100\% \times 50\%} = \sim 70.7\% \)
RQ1: To what extent does resampling in OOB and UOB help dealing with class imbalance online?

• Comparison of OOB and UOB against OB.
• Ensembles use 50 Hoeffding trees.
Both OOB and UOB, and particularly UOB, greatly improve minority class recall over OB.

Resampling is important in improving UOB and OOB's performance. UOB tends to outperform OOB.

Gmean is also better.

<table>
<thead>
<tr>
<th>Data Distribute</th>
<th>IR</th>
<th>OOB</th>
<th>UOB</th>
<th>OB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30%</td>
<td>0.970±0.002</td>
<td>0.973±0.001</td>
<td>0.969±0.001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.981±0.003</td>
<td>0.964±0.002</td>
<td>0.964±0.004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00510)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.912±0.007</td>
<td>0.936±0.005</td>
<td>0.905±0.007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.840±0.000</td>
<td>0.917±0.013</td>
<td>0.876±0.019</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td>safe</td>
<td>30%</td>
<td>0.636±0.013</td>
<td>0.774±0.007</td>
<td>0.462±0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.577±0.007</td>
<td>0.749±0.005</td>
<td>0.429±0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.424±0.000</td>
<td>0.665±0.017</td>
<td>0.235±0.008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.225±0.000</td>
<td>0.519±0.021</td>
<td>0.008±0.009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td>borderline</td>
<td>30%</td>
<td>0.197±0.016</td>
<td>0.662±0.033</td>
<td>0.033±0.004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.395±0.015</td>
<td>0.755±0.014</td>
<td>0.142±0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.195±0.024</td>
<td>0.699±0.014</td>
<td>0.195±0.024</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.310±0.010</td>
<td>0.519±0.021</td>
<td>0.008±0.009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td>rare/outlier</td>
<td>30%</td>
<td>0.045±0.009</td>
<td>0.446±0.041</td>
<td>0.000±0.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.395±0.015</td>
<td>0.755±0.014</td>
<td>0.142±0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.195±0.024</td>
<td>0.699±0.014</td>
<td>0.195±0.024</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.310±0.010</td>
<td>0.519±0.021</td>
<td>0.008±0.009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td>Gearbox</td>
<td></td>
<td>0.045±0.009</td>
<td>0.446±0.041</td>
<td>0.000±0.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
<tr>
<td>Smart Building</td>
<td></td>
<td>0.430±0.004</td>
<td>0.764±0.011</td>
<td>0.234±0.103</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.00000)</td>
<td>(0.00000)</td>
<td>(0.00000)</td>
</tr>
</tbody>
</table>
RQ2: How do OOB and UOB perform in comparison with previous approaches?

- Comparison of OOB and UOB against RLSACP and WOS-ELM.
- OOB and UOB use 50 MLPs or 1 MLP.
- RLSACP = Recursive Least Square Adaptive Cost Perceptron.


- WOS-ELM = Weighted Online Sequential Extreme Learning Machine.

RQ2: How do OOB and UOB perform in comparison with previous approaches?

Minority class recall at the last step of the training

<table>
<thead>
<tr>
<th>Data Distribute</th>
<th>IR</th>
<th>OOB</th>
<th>UOB</th>
<th>RLSACP</th>
<th>WOS-ELM</th>
</tr>
</thead>
<tbody>
<tr>
<td>safe</td>
<td>30%</td>
<td>0.973±0.004 (0.00000)</td>
<td>0.986±0.001</td>
<td>0.490±0.344 (0.00000)</td>
<td>0.332±0.466 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.979±0.003 (0.00000)</td>
<td>0.992±0.007</td>
<td>0.551±0.373 (0.00000)</td>
<td>0.283±0.443 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.923±0.005 (0.00620)</td>
<td>0.900±0.048</td>
<td>0.593±0.302 (0.00000)</td>
<td>0.586±0.466 (0.97450)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.880±0.000 (0.00000)</td>
<td>0.741±0.120</td>
<td>0.027±0.025 (0.00000)</td>
<td>0.414±0.449 (0.02530)</td>
</tr>
<tr>
<td>borderline</td>
<td>30%</td>
<td>0.488±0.016 (0.00000)</td>
<td>0.828±0.036</td>
<td>0.512±0.020 (0.00000)</td>
<td>0.194±0.286 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.369±0.017 (0.00000)</td>
<td>0.854±0.055</td>
<td>0.535±0.028 (0.00000)</td>
<td>0.520±0.341 (0.00002)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.293±0.011 (0.00000)</td>
<td>0.806±0.106</td>
<td>0.481±0.064 (0.00000)</td>
<td>0.296±0.398 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.015±0.008 (0.00000)</td>
<td>0.456±0.212</td>
<td>0.039±0.017 (0.00000)</td>
<td>0.417±0.348 (0.21640)</td>
</tr>
<tr>
<td>rare/ outlier</td>
<td>30%</td>
<td>0.196±0.019 (0.00000)</td>
<td>0.727±0.047</td>
<td>0.493±0.115 (0.00000)</td>
<td>0.559±0.149 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.370±0.011 (0.00000)</td>
<td>0.853±0.052</td>
<td>0.468±0.015 (0.00000)</td>
<td>0.234±0.266 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.390±0.015 (0.00000)</td>
<td>0.839±0.088</td>
<td>0.521±0.144 (0.00000)</td>
<td>0.363±0.419 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.181±0.004 (0.00000)</td>
<td>0.479±0.200</td>
<td>0.111±0.030 (0.00000)</td>
<td>0.164±0.270 (0.00000)</td>
</tr>
<tr>
<td>Gearbox</td>
<td></td>
<td>0.008±0.005 (0.00000)</td>
<td>0.697±0.110</td>
<td>0.042±0.023 (0.00000)</td>
<td>0.888±0.022 (0.00000)</td>
</tr>
<tr>
<td>Smart Building</td>
<td></td>
<td>0.065±0.014 (0.00000)</td>
<td>0.552±0.075</td>
<td>0.161±0.280 (0.00000)</td>
<td>0.484±0.011 (0.00000)</td>
</tr>
</tbody>
</table>

UOB obtains better minority-class recall
RQ2: How do OOB and UOB perform in comparison with previous approaches?

G-mean at the last step of the training

<table>
<thead>
<tr>
<th>Data Distribute</th>
<th>IR</th>
<th>OOB</th>
<th>UOB</th>
<th>RLSACP</th>
<th>WOS-ELM</th>
</tr>
</thead>
<tbody>
<tr>
<td>safe</td>
<td>30%</td>
<td>0.972±0.001</td>
<td>0.926±0.007 (0.00000)</td>
<td>0.493±0.345 (0.00000)</td>
<td>0.065±0.066 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.970±0.001</td>
<td>0.907±0.010 (0.00000)</td>
<td>0.548±0.365 (0.00000)</td>
<td>0.036±0.077 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.957±0.002</td>
<td>0.842±0.025 (0.00000)</td>
<td>0.593±0.304 (0.00000)</td>
<td>0.146±0.135 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.933±0.000</td>
<td>0.776±0.040 (0.00000)</td>
<td>0.123±0.105 (0.00000)</td>
<td>0.160±0.181 (0.00000)</td>
</tr>
<tr>
<td>borderline</td>
<td>30%</td>
<td>0.586±0.007</td>
<td>0.515±0.022 (0.00000)</td>
<td>0.515±0.079 (0.00000)</td>
<td>0.231±0.137 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.537±0.010</td>
<td>0.426±0.050 (0.00000)</td>
<td>0.475±0.086 (0.00000)</td>
<td>0.348±0.138 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.500±0.009</td>
<td>0.374±0.096 (0.00000)</td>
<td>0.467±0.123 (0.02940)</td>
<td>0.189±0.113 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.104±0.061</td>
<td>0.447±0.060 (0.00000)</td>
<td>0.183±0.055 (0.00000)</td>
<td>0.306±0.172 (0.00000)</td>
</tr>
<tr>
<td>rare/outlier</td>
<td>30%</td>
<td>0.399±0.016</td>
<td>0.463±0.026 (0.00000)</td>
<td>0.513±0.021 (0.00000)</td>
<td>0.476±0.056 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.561±0.007</td>
<td>0.425±0.060 (0.00000)</td>
<td>0.482±0.093 (0.00000)</td>
<td>0.254±0.228 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.598±0.011</td>
<td>0.447±0.094 (0.00000)</td>
<td>0.516±0.153 (0.09300)</td>
<td>0.163±0.170 (0.00000)</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.416±0.004</td>
<td>0.450±0.081 (0.00000)</td>
<td>0.316±0.047 (0.00000)</td>
<td>0.162±0.208 (0.00000)</td>
</tr>
<tr>
<td>Gearbox</td>
<td></td>
<td>0.077±0.047</td>
<td>0.459±0.055 (0.00000)</td>
<td>0.189±0.063 (0.00000)</td>
<td>0.289±0.022 (0.00000)</td>
</tr>
<tr>
<td>Smart Building</td>
<td></td>
<td>0.243±0.027</td>
<td>0.485±0.020 (0.00000)</td>
<td>0.220±0.081 (0.00033)</td>
<td>0.527±0.004 (0.00000)</td>
</tr>
</tbody>
</table>

With RTs: UOB is more aggressive, reducing majority class recall and achieving both better minority class recall and G-mean.

With MLPs: majority class recall is affected by 1 epoch learning. UOB pushes it even lower, leading to worse G-mean.
Analysis in Dynamic Data Streams

Research questions:

1. How does the time-decayed metric used in OOB and UOB help handling imbalance change?

2. How do OOB and UOB perform in comparison with previous methods under dynamic scenarios?
Experimental Setup

Data sets:

- 4 two-class data stream created based on Gaussian distribution.
- 4 gearbox fault detection.
- 4 sensor contaminant detection in smart buildings.
  - First 500 examples have 10% imbalance rate, (+) class is minority.
  - A change happens at time step 501:
    - High severity: (-) class becomes minority with 10% imbalance rate.
    - Low severity: data become balanced.
    - Abrupt: change completes in 1 time step.
    - Gradual: change takes 300 time steps to complete.

Performance measures:

- Prequential minority class recall and G-mean, reset after time steps 500 [and 800].
- Wilcoxon sign-rank tests with holm-bonferroni corrections for average performance across time.
RQ1: How does the time-decayed metric used in OOB and UOB help handling imbalance change?

Comparisons:
• OOB and UOB are compared with traditional methods of updating the class size (OOBtr and UOBtr).
• OOBtr and UOBtr consider all examples so far equally when determining the class size, i.e., they use no decay function.

Base learners:
• All ensembles use 50 Hoeffding trees.
RQ1: How does the time-decayed metric used in OOB and UOB help handling imbalance change?

All approaches improve recall on a change from minority to majority. OOB is more robust to changes than UOB.

Time-decayed metric is important.
RQ2: How do OOB and UOB perform in comparison with previous methods under dynamic scenarios?

Comparisons:

• OOB and UOB are compared with RLSACP and WOS-ELM.

Base learners:

• OOB and UOB use 50 MLPs or 1 MLP.
RQ2: How do OOB and UOB perform in comparison with previous methods under dynamic scenarios?

- OOB or UOB achieve better results

<table>
<thead>
<tr>
<th>Data</th>
<th>Change</th>
<th>OOB</th>
<th>UOB</th>
<th>RLSACP</th>
<th>WOS-ELM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>H A</td>
<td>0.532±0.034</td>
<td>0.019±0.013</td>
<td>(0.00000)</td>
<td>0.366±0.096</td>
</tr>
<tr>
<td></td>
<td>H G</td>
<td>0.360±0.002</td>
<td>0.359±0.011</td>
<td>(0.00440)</td>
<td>0.388±0.089</td>
</tr>
<tr>
<td></td>
<td>L A</td>
<td>0.795±0.003</td>
<td>0.521±0.025</td>
<td>(0.00000)</td>
<td>0.439±0.210</td>
</tr>
<tr>
<td></td>
<td>L G</td>
<td>0.811±0.001</td>
<td>0.758±0.015</td>
<td>(0.00000)</td>
<td>0.613±0.118</td>
</tr>
<tr>
<td>Gearbox</td>
<td>H A</td>
<td>0.293±0.009</td>
<td>0.201±0.032</td>
<td>(0.00000)</td>
<td>0.438±0.012</td>
</tr>
<tr>
<td></td>
<td>H G</td>
<td>0.000±0.000</td>
<td>0.364±0.119</td>
<td>(0.00000)</td>
<td>0.140±0.126</td>
</tr>
<tr>
<td></td>
<td>L A</td>
<td>0.452±0.014</td>
<td>0.479±0.012</td>
<td>(0.00000)</td>
<td>0.378±0.031</td>
</tr>
<tr>
<td></td>
<td>L G</td>
<td>0.408±0.032</td>
<td>0.432±0.023</td>
<td>(0.00000)</td>
<td>0.055±0.095</td>
</tr>
<tr>
<td>Smart</td>
<td>H A</td>
<td>0.276±0.028</td>
<td>0.261±0.052</td>
<td>(0.04740)</td>
<td>0.135±0.156</td>
</tr>
<tr>
<td>Building</td>
<td>H G</td>
<td>0.000±0.000</td>
<td>0.140±0.091</td>
<td>(0.12090)</td>
<td>0.157±0.188</td>
</tr>
<tr>
<td></td>
<td>L A</td>
<td>0.451±0.019</td>
<td>0.435±0.016</td>
<td>(0.00000)</td>
<td>0.228±0.144</td>
</tr>
<tr>
<td></td>
<td>L G</td>
<td>0.443±0.023</td>
<td>0.485±0.018</td>
<td>(0.00000)</td>
<td>0.080±0.078</td>
</tr>
</tbody>
</table>

Either OOB or UOB achieve better results in most cases.

Similar results are achieved when OOB and UOB use a single MLP, suggesting that resampling plays a key role.

The proposed approaches are competitive in comparison to previous approaches.
Ensemble of OOB and UOB

• UOB was better than OOB specially during static periods.
• OOB can adapt better to changes from majority to minority.
• Can we combine the strengths of OOB and UOB?
  • Ensemble of OOB and UOB (WEOB): the strategy with highest smoothed time-decayed G-mean is chosen for making predictions.
  • WEOB performs better than, or between OOB and UOB.
    • It achieved better performance than OOB and better robustness to changes than UOB.
Summary of Resampling-Based Ensembles for Class Imbalance Learning

- [Time-decayed] resampling played an important role in OOB and UOB's ability to deal with [changing] imbalance rates.
- OOB and UOB were competitive against previous approaches, both in static and dynamic settings.
- Data distributions, imbalance rates and base classifiers have significant impact on G-mean, with data distribution being the most influential factor. Base learners are also influential.
- UOB was better in static and OOB in dynamic settings.
- WEOB can combine the advantages of OOB and UOB.
Online Class Imbalance Learning Framework

- Data Stream
  - 1. Class Imbalance Detector
    - Output
    - Imbalance Status
  - 2. Concept Drift Detector
    - Output
    - Drift for each class
  - 3. Online Learner
Concept Drift Detection Under Class Imbalance

- Typical concept drift detectors monitor accuracy.
- Would that be adequate for class-imbalanced scenarios?
Experimental Study

Objectives:

• Check the suitability of accuracy as a measure to be monitored for drift detection.

Comparisons:

• Time-decayed accuracy vs recall on minority class.

Base learners:

• Online bagging with 50 DTs
• Online bagging with 50 NB
Data Sets

SEA Concepts
\[ \alpha_1 + \alpha_2 \leq \theta \rightarrow \text{class 0} \]
\[ \alpha_1 + \alpha_2 > \theta \rightarrow \text{class 1 (minority)} \]

STAGGER
Boolean equations involving size (S, M, L), shape (circle, triangle, rectangle) and colour (R, G, B).
False \rightarrow \text{class 0}
True \rightarrow \text{class 1 (minority)}

iNemo
Multi-sensing platform (accelerometers, gyroscopes and magnetometers with pressure and temperature). An offset \( \theta \) is added to the gyroscope signal to simulate faults.
Non-faulty \rightarrow \text{class 0}
Faulty \rightarrow \text{class 1 (minority)}
Data Sets

Concept Drift

<table>
<thead>
<tr>
<th>Concept</th>
<th>SEA</th>
<th>STAGGER</th>
<th>iNemo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
<td>$\theta = 13$</td>
<td>size=small &amp; color=red</td>
<td>$\theta = 500$</td>
</tr>
<tr>
<td>New Low Severity</td>
<td>$\theta = 10$</td>
<td>(size=small &amp; color=red) | (color=green &amp; shape=square)</td>
<td>$\theta = 500$ | $\theta = -500$</td>
</tr>
<tr>
<td>New High Severity</td>
<td>$\theta = 7$</td>
<td>color=green &amp; shape=square</td>
<td>$\theta = -500$</td>
</tr>
</tbody>
</table>

Class Imbalance Status

<table>
<thead>
<tr>
<th></th>
<th>$p_{c0}$</th>
<th>$p_{c1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>New Low Severity</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>New High Severity</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Data stream size: 1000 examples
Abrupt change at 501
Time-Decayed Accuracy Over Time

• Concept drift has similar pattern to no-change.
• Class-imbalance change leads to a reduction in accuracy.
• We want to detect concept drifts.

SEA: time-decayed overall accuracy under the scenarios of high severity changes produced by DT-based OB.

Accuracy is not adequate for detecting concept drift in class-imbalance environments!
Time-Decayed Recall on Minority Class

- Changes with concept drift cause drops in recall.
- Changes in class imbalance don't.

This allows us to treat these two types of change separately!
Drift Detection Method for Online Class Imbalance Learning (DDM-OCI)

1) Monitors minority-class recall $p_i$ and standard deviation $s_i$.
2) Record $p_{\text{max}}$ and $s_{\text{max}}$ to remember when $p_i + s_i$ reaches its maximum, and check the following conditions:
   • Warning level $p_i - s_i \leq p_{\text{max}} - 2s_{\text{max}}$ is reached at time $t_w$. A potential drift is considered to start from this moment. Examples coming after $t_w$ are stored.
   • Drift level $p_i - s_i \leq p_{\text{max}} - 3s_{\text{max}}$ is reached at time $t_d$. The online model and all the recorded values are reset. A new model is induced using the examples stored between $t_w$ and $t_d$.

Experimental Study

Objectives:

• Check the suitability of DDM-OCI.

Comparisons:

• DDM-OCI vs Drift Detection Method (DDM).


Base learners:

• Online bagging with 50 DTs
• Online bagging with 50 NB
Results

No drift detection performs the worse.

DDM-OCI reacts faster.

DDM-OCI suffers from false alarms.

Results with DTs.
Number of Concept Drift Detections

- Only one concept drift happens in cases 4-7.
- DDM sometimes fails to detect concept drifts.
- DDM-OCI has false alarms.
  - False alarms are not necessarily a big problem for approaches such as DDD.
- Detecting drifts early is more important.

<table>
<thead>
<tr>
<th></th>
<th>SEA</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case</strong></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td><strong>DDM</strong></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>DDM-OCI</strong></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td><strong>STAGGER</strong></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>iNemo</strong></td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

For results with DTs.
Summary of Drift Detection Method for Online Class Imbalance

- Accuracy is not an ideal measure for drift detection.
- Recall of minority class allows us to distinguish from changes in class imbalance and concept drifts.
- DDM-OCI uses recall of minority class instead of accuracy to detect changes.
- It usually detects concept drifts earlier than DDM, even though it leads to more false alarms.
Outline

• Introduction
• What types of change exist?
• How to evaluate approaches under changes?
• When, why and how ensembles can help dealing with changes?
• How to achieve robustness to different types of change?
• What to do if we actually have "little data"?
• How to handle class imbalance?
• Future directions
Future Directions

• Online class imbalance learning

• Semi-supervised online learning


• Scalable memories

References


Thank you!

Sharing Data and Models in Software Engineering
Tim Menzies, Ekrem Kocaguneli, Leandro Minku, Fayola Peters and Burak Turhan