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.

- 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

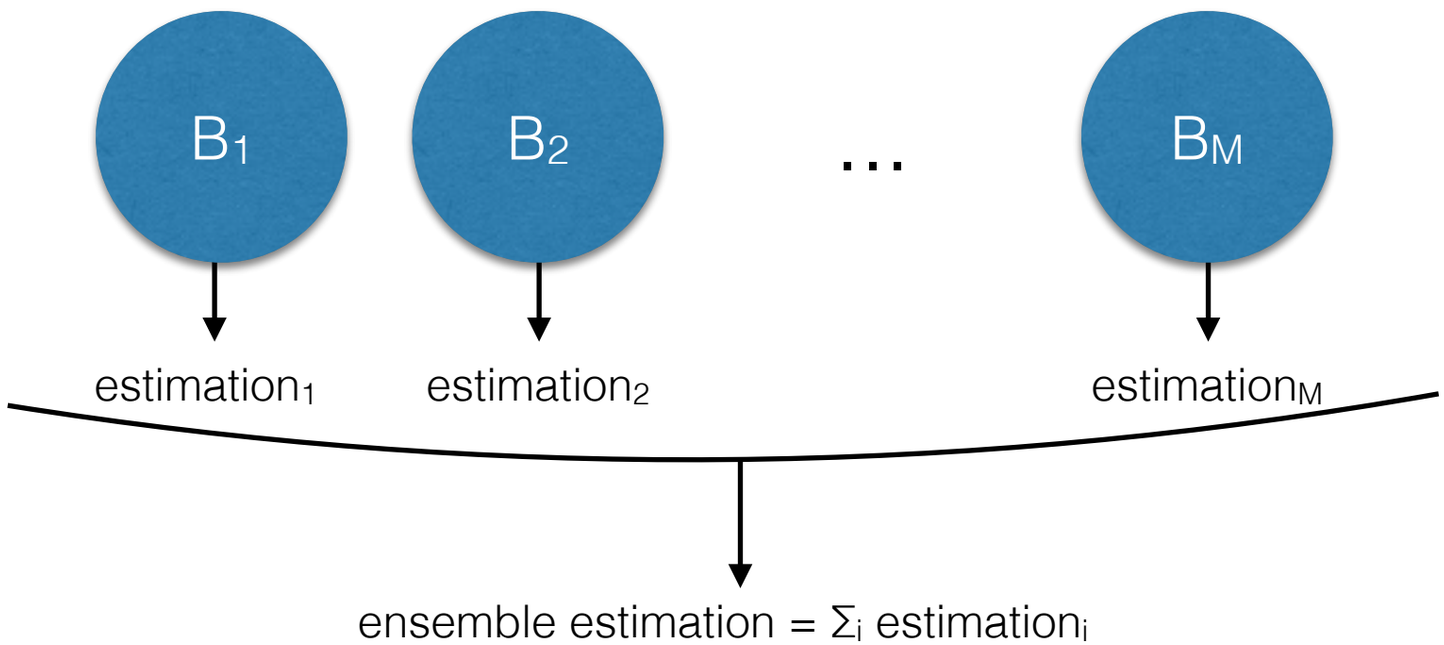
An ideal online learning algorithm for changing environments should:

- 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

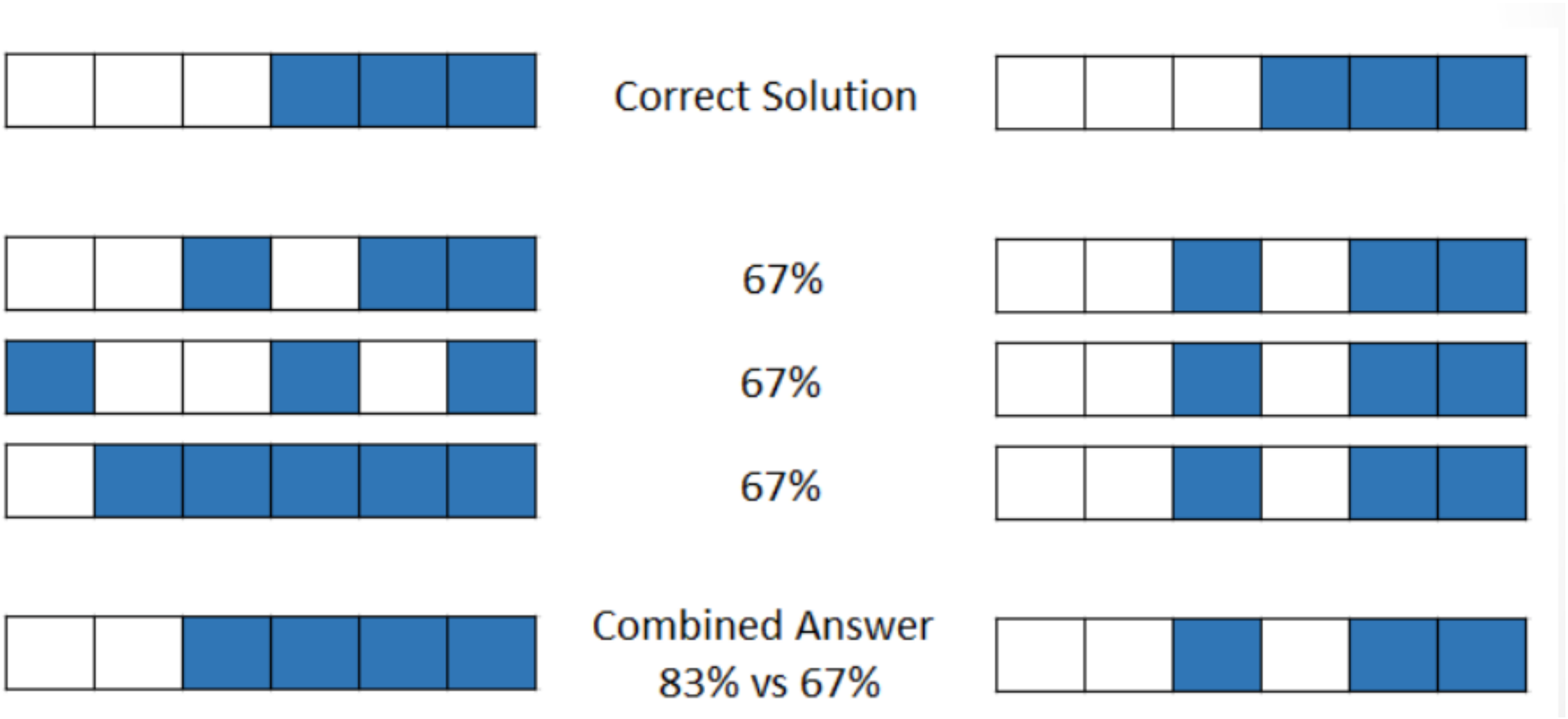
- Ensembles: sets of learning machines grouped together.
- Aim: to improve predictive performance.
- Key: diversity, i.e., models give different errors on the same examples.



[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 $p(\boldsymbol{x}, w)$:

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

or

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

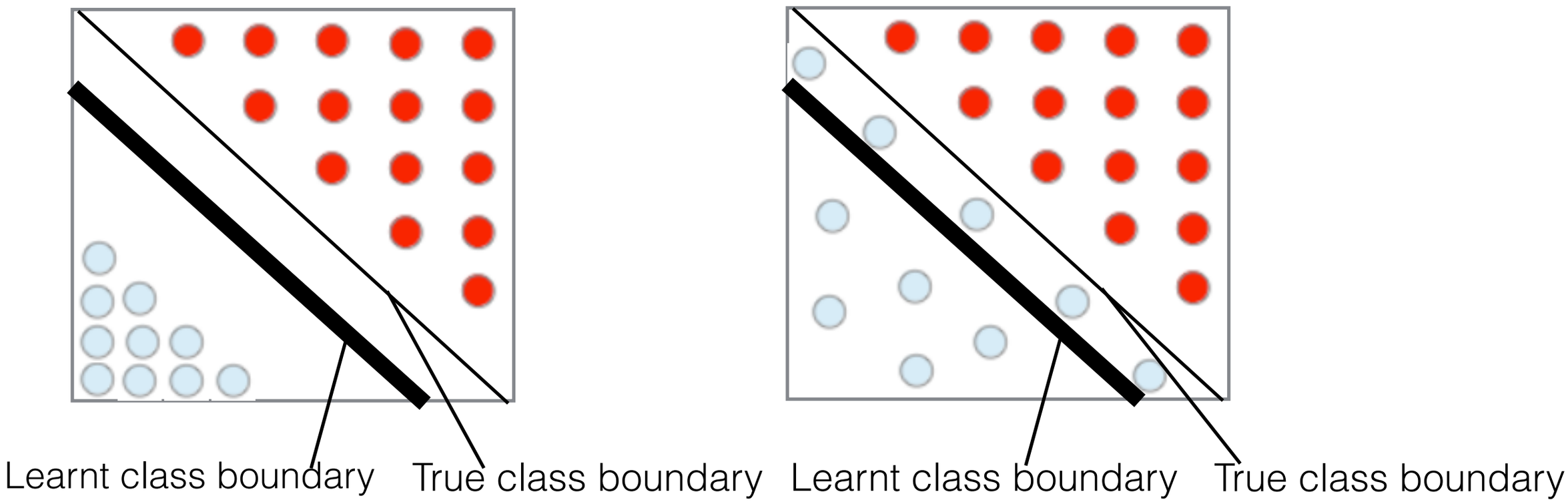
L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010

Why Adopting this Definition?

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

- 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!



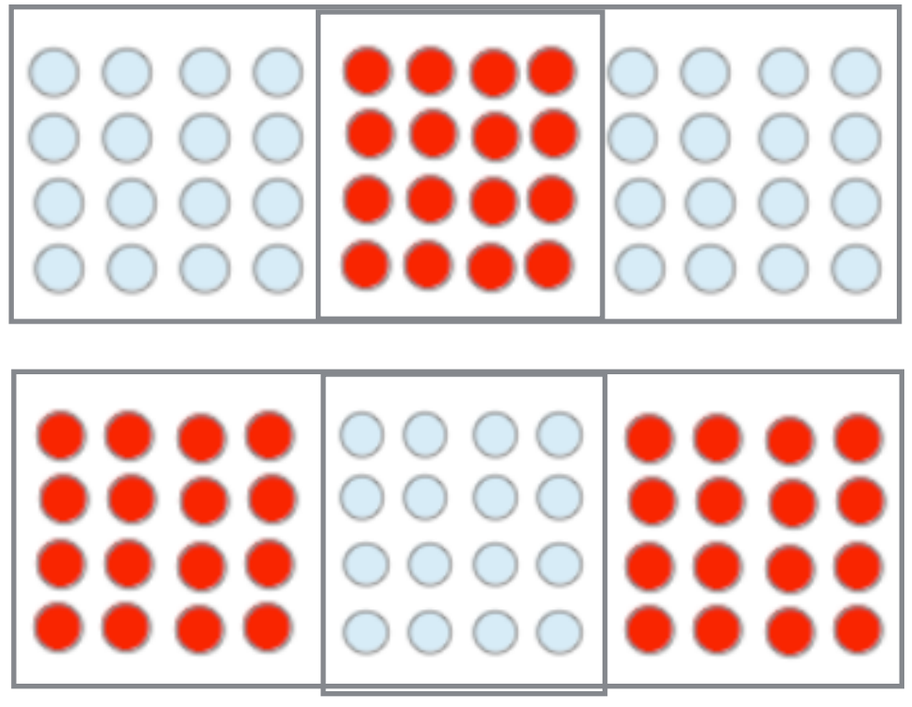
Characterising Concept Drifts in Isolation

Drift in Isolation	Severity	Class
		Feature
	Speed	

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

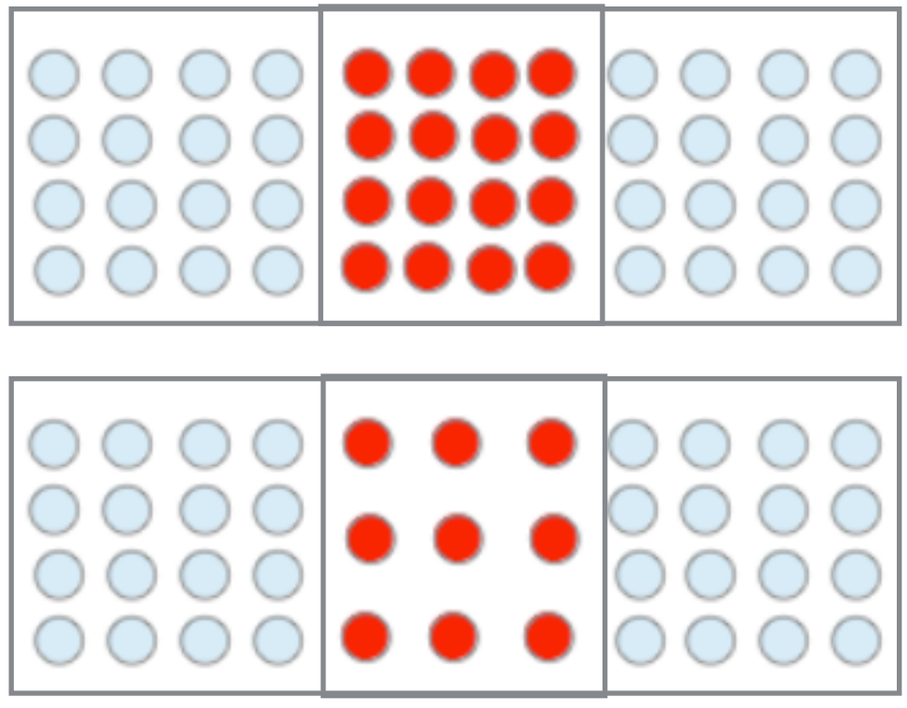
Characterising Concept Drifts in Isolation

Drift in Isolation	Severity	Class
		Feature
	Speed	

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.



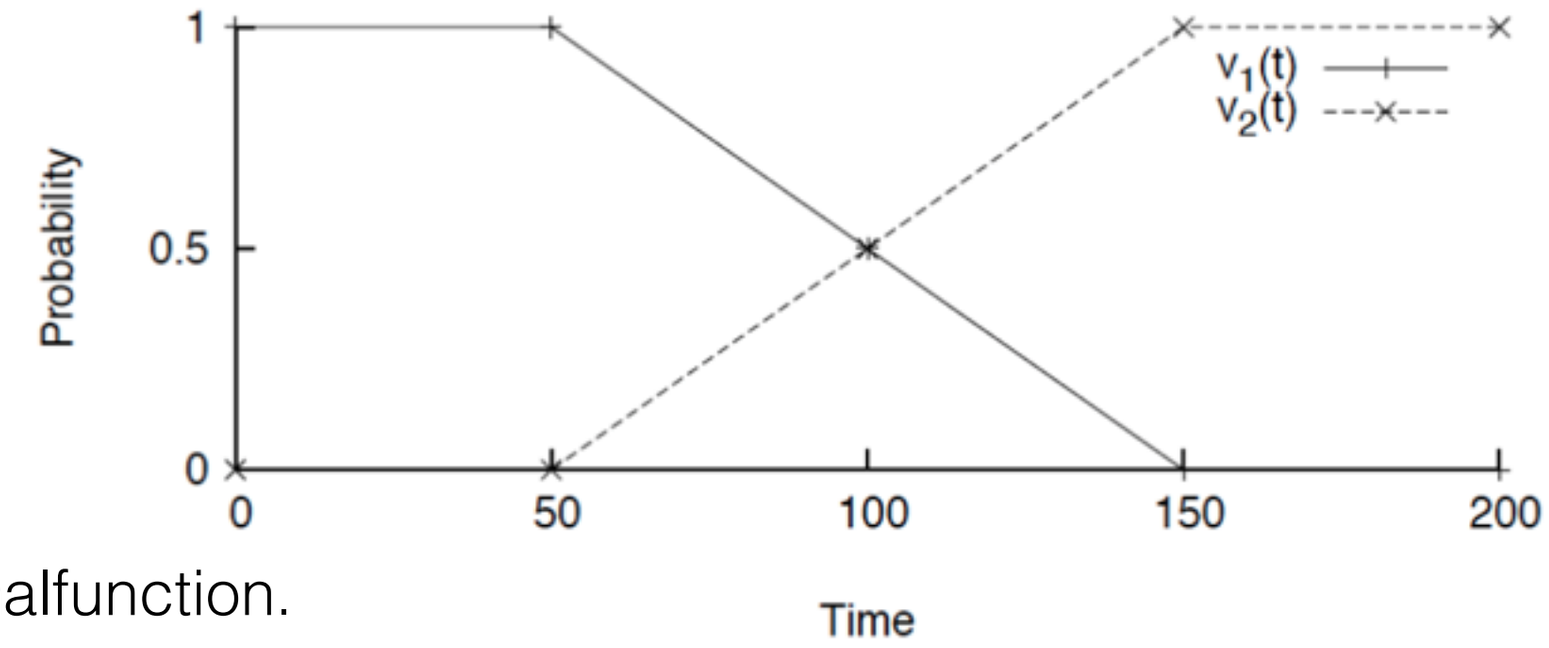
L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010

Characterising Concept Drifts in Isolation

Drift in Isolation	Severity	Class
		Feature
	Speed	

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

Sequences of Drifts	Frequency
	Recurrence
	Predictability

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

Sequences of Drifts	Frequency
	Recurrence
	Predictability

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.

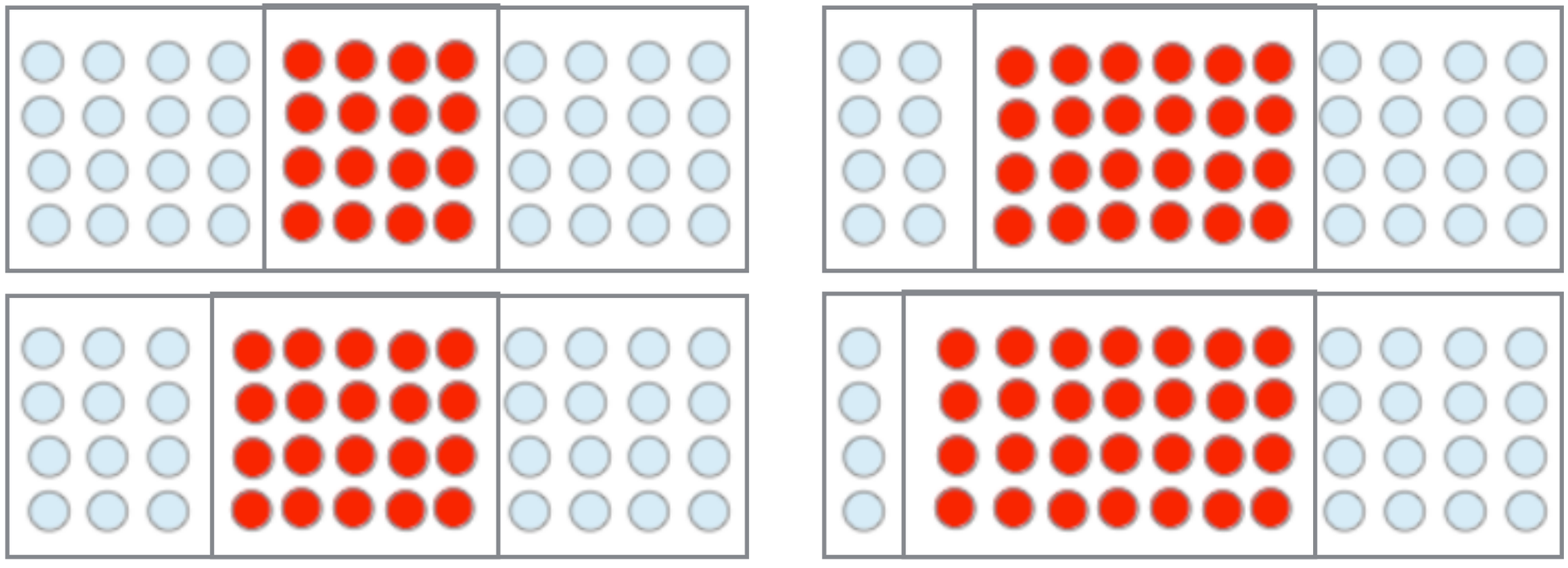
L. L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, 22(5):730-742, 2010

Characterising Sequences of Concept Drifts

Predictability: whether the sequence of changes can be learnt to predict future changes.

Example: people buying more and more selfie sticks.

Sequences of Drifts	Frequency
	Recurrence
	Predictability



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

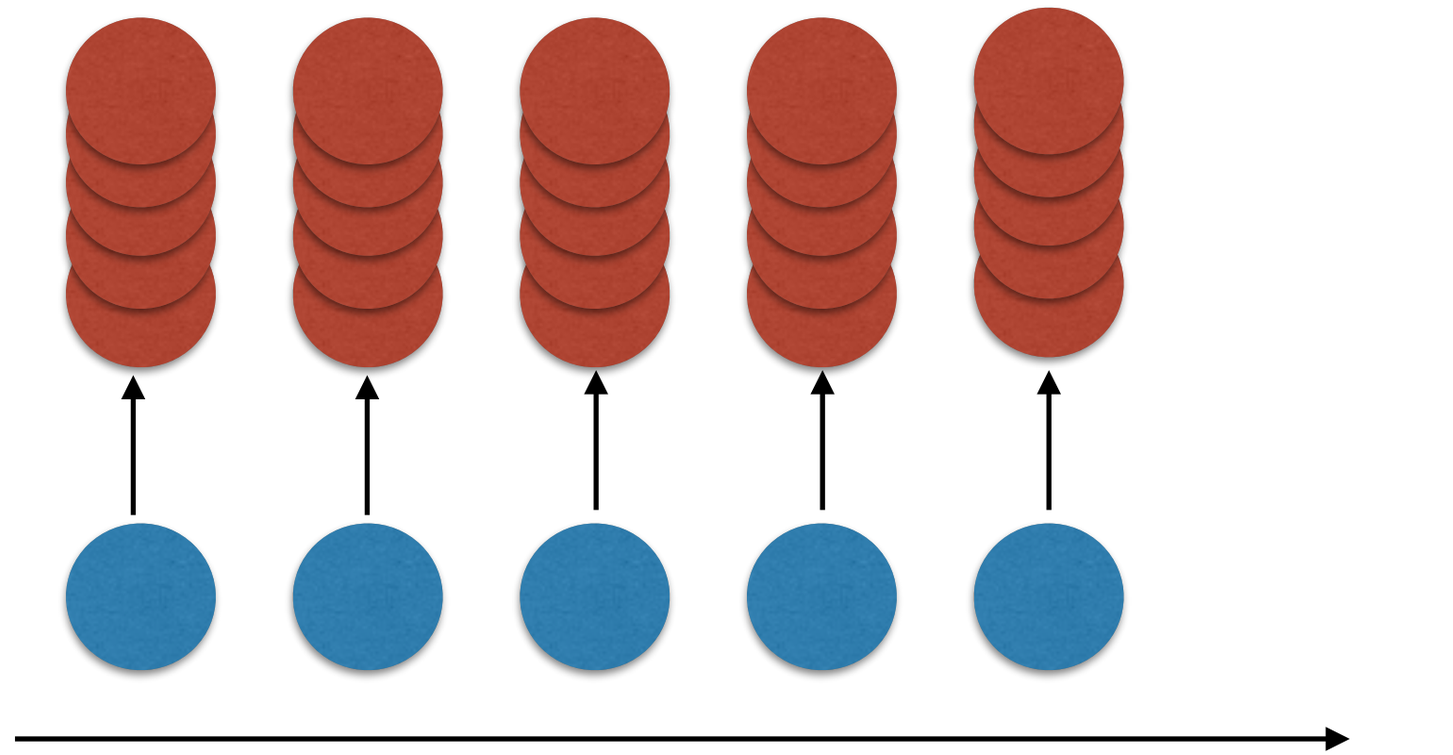
- Introduction
- What types of change exist?
- How to evaluate approaches under changes?
- When, why and how ensembles can help dealing with changes?
- How to achieve robustness to different types of change?
- How to handle class imbalance?
- What to do if we actually have "little data"?
- Future directions

Data Sets

- Artificial vs real world data:
  - http://www.cs.bham.ac.uk/~minkull/opensource/ArtificialConceptDriftDataSets.zip
- Test on different types of concept drift.
  - http://en.wikipedia.org/wiki/Concept_drift

Computing Performance

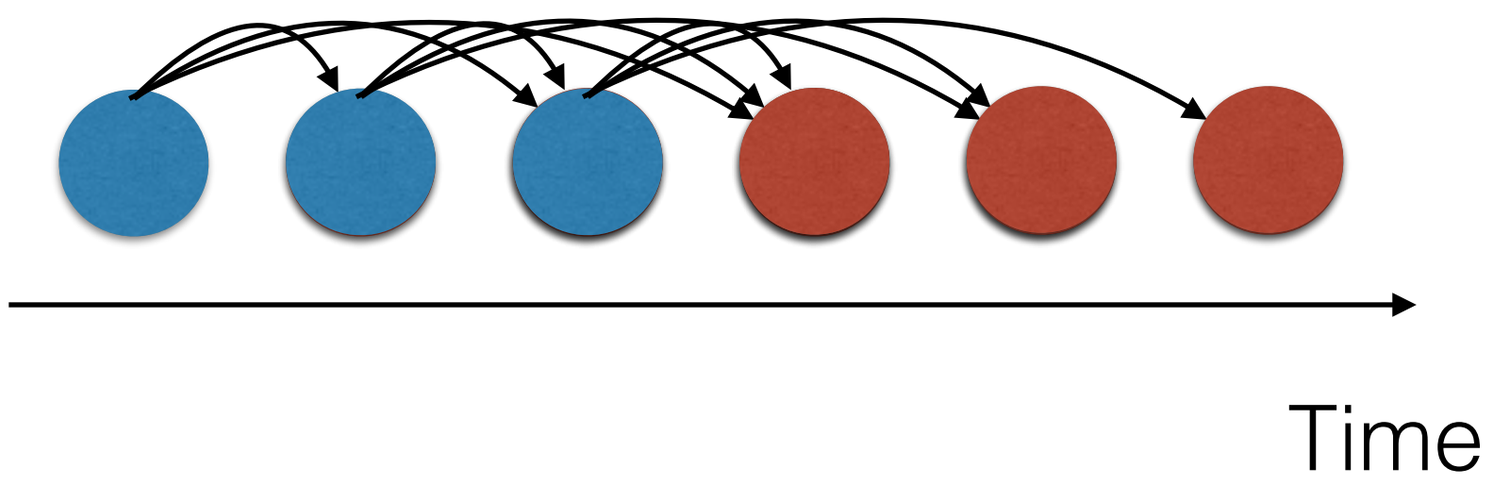
- Performance on a separate test set.



Time

Computing Performance

- Performance on the next *X* examples.

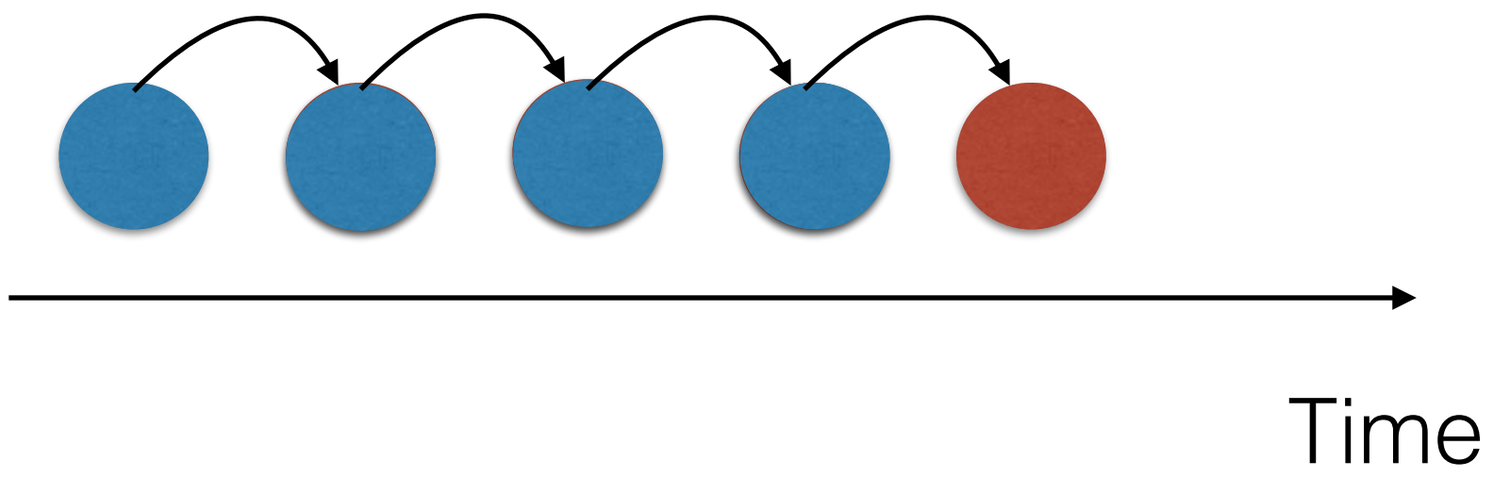


Computing Performance

- 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:

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

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

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

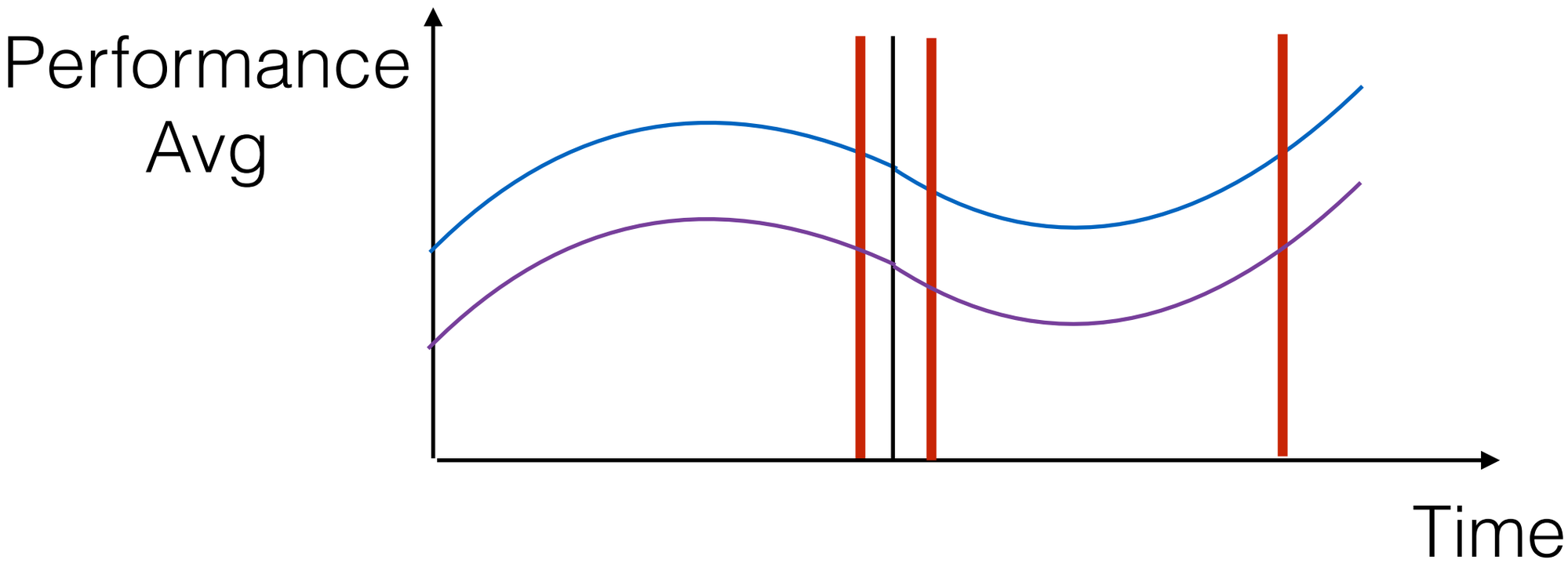
J.Gama, R.Sebastiao, P.P.Rodrigues. Issues in Evaluation of Stream Learning Algorithms. Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, 329338, 2009.

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!

BUT MIIIINECRAFT!

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 CYAN JELLY BEANS AND ACNE (P > 0.05).

WE FOUND A LINK BETWEEN GREEN JELLY BEANS AND ACNE (P < 0.05).

WHOA!

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

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

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

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

WE FOUND NO LINK BETWEEN PEACH 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 PURPLE 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 PINK 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 SALMON JELLY BEANS AND ACNE (P > 0.05).

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

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

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 CYAN JELLY BEANS AND ACNE (P > 0.05).

WE FOUND A LINK BETWEEN GREEN JELLY BEANS AND ACNE (P < 0.05).

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

News

GREEN JELLY BEANS LINKED TO ACNE!

95% CONFIDENCE

ONLY 5% CHANCE OF COINCIDENCE!

SCIENTISTS...

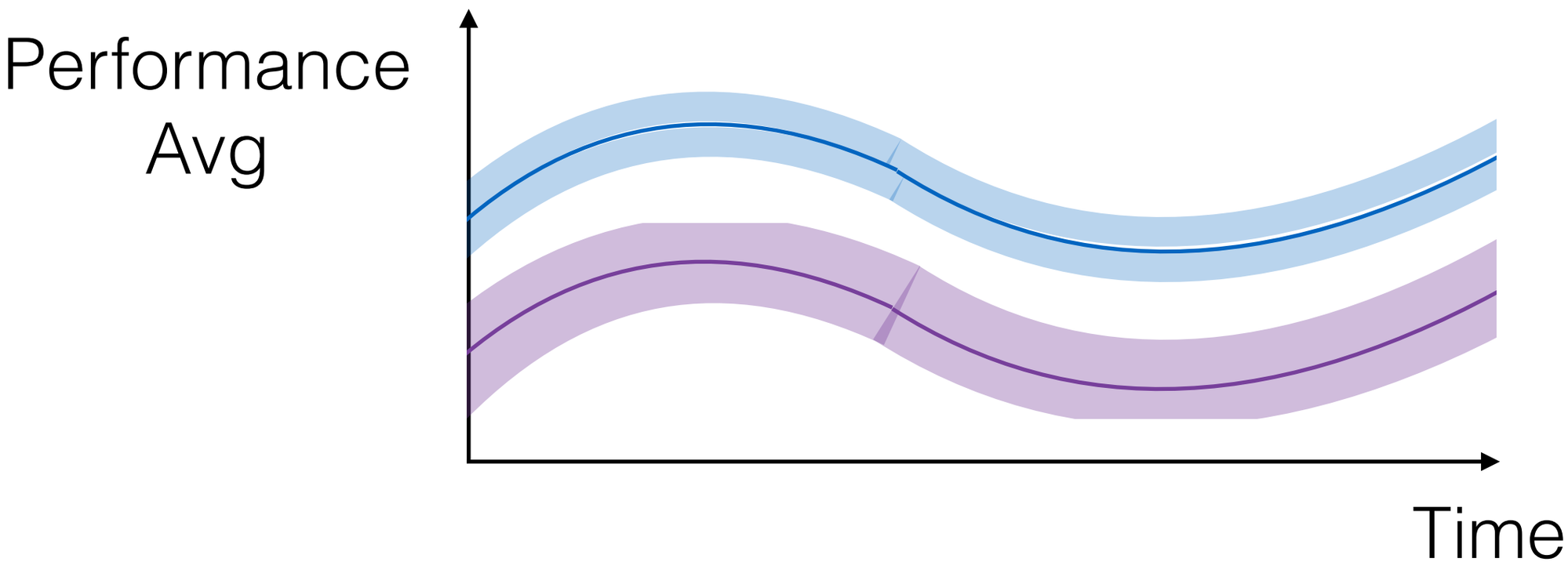
Source: http://xkcd.com/882/

Are the Performances of Different Approaches Really Different?

Specially when our predictors are stochastic...

Confidence intervals throughout the learning.

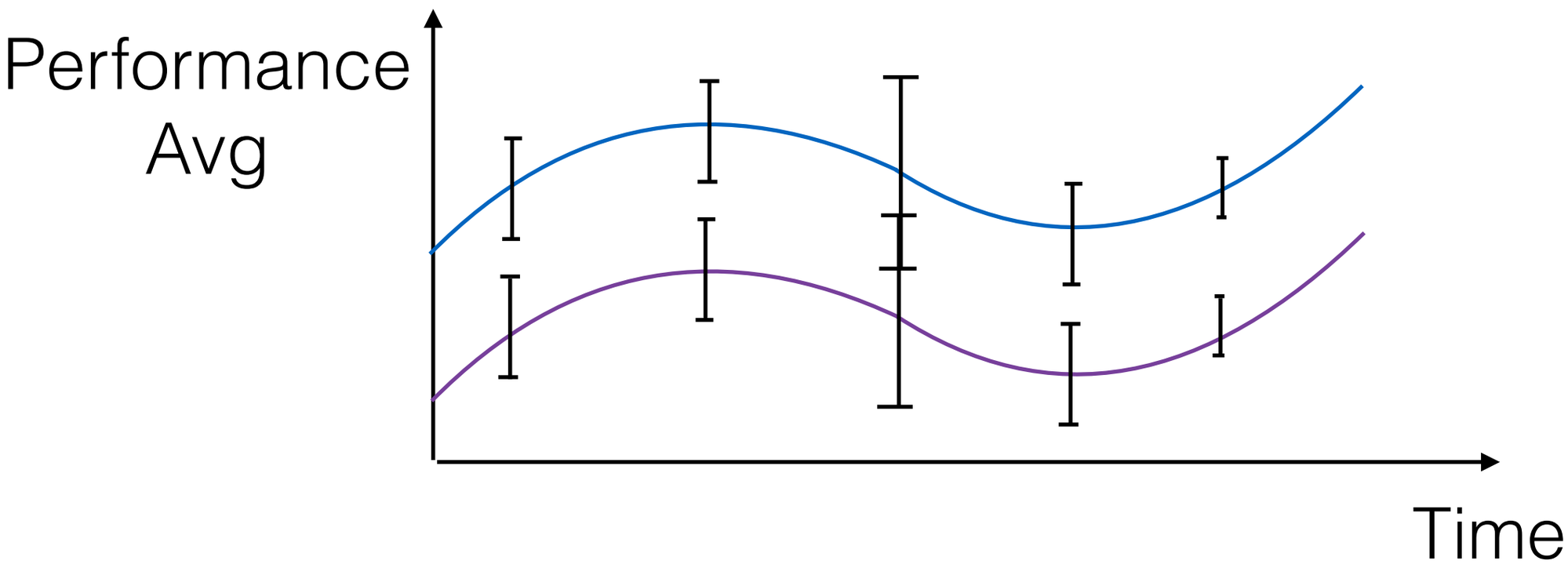
- Given a set of estimations for the same example, what interval would contain estimation values that would not be unusual?



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.

J. Demsar. Statistical comparisons of classifiers over multiple data sets. JMLR, 7:1–30, 2006.

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.

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 $K \sim Poisson(\lambda)$
  update f_m K times
end for

Changing sampling rate will lead to different levels of diversity.

Hoeffding Trees.

Poisson distribution to set sample.

N. C. Oza and S. Russell, "Experimental comparisons of online and batch versions of bagging and boosting," in Proc. of ACM SIGKDD, San Francisco, 2001, pp. 359–364.

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

L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy", Machine Learning, 51:181–207, 2003.

Data Sets

Experiments on 6 x 3 x 3 artificial data sets.

Probl.	Equation	Fixed Values	Before→After Drift	Sev.
Circle	$(x-a)^2+(y-b)^2 \leq r^2$	$a=.5$ $b=.5$	$r=.2 \rightarrow .3$	16%
			$r=.2 \rightarrow .4$	38%
			$r=.2 \rightarrow .5$	66%
SineV	$y \leq a\,sin(bx+c)+d$	$a=1$ $b=1$ $c=0$	$d=-2 \rightarrow 1$	15%
			$d=-5 \rightarrow 4$	45%
			$d=-8 \rightarrow 7$	75%
SineH	$y \leq a\,sin(bx+c)+d$	$a=5$ $d=5$ $b=1$	$c=0 \rightarrow -\pi/4$	36%
			$c=0 \rightarrow -\pi/2$	57%
			$c=0 \rightarrow -\pi$	80%
Line	$y \leq -a_0+a_1x_1$	$a_1=.1$	$a_0=-.4 \rightarrow -.55$	15%
			$a_0=-.25 \rightarrow -.7$	45%
			$a_0=-.1 \rightarrow -.8$	70%
Plane	$y \leq -a_0+a_1x_1+a_2x_2$	$a_1=.1$ $a_2=.1$	$a_0=-2 \rightarrow -2.7$	14%
			$a_0=-1 \rightarrow -3.2$	44%
			$a_0=-.7 \rightarrow -4.4$	74%
Bool	$(color\ eq_1\ a\ op_1\ shape\ eq_2\ b)\ op_2\ size\ eq_3\ c$	$c=S \vee M \vee L$ $op_2 \wedge$ $eq_{1,2,3}=$	$a=R, op_1 \wedge b=R \rightarrow R \vee T$	11%
			$a=R, b=R, op_1 \wedge \rightarrow \vee$	44%
			$a=R \rightarrow R \vee G, b=R \rightarrow R \vee T, op_1 \wedge \rightarrow \vee$	67%

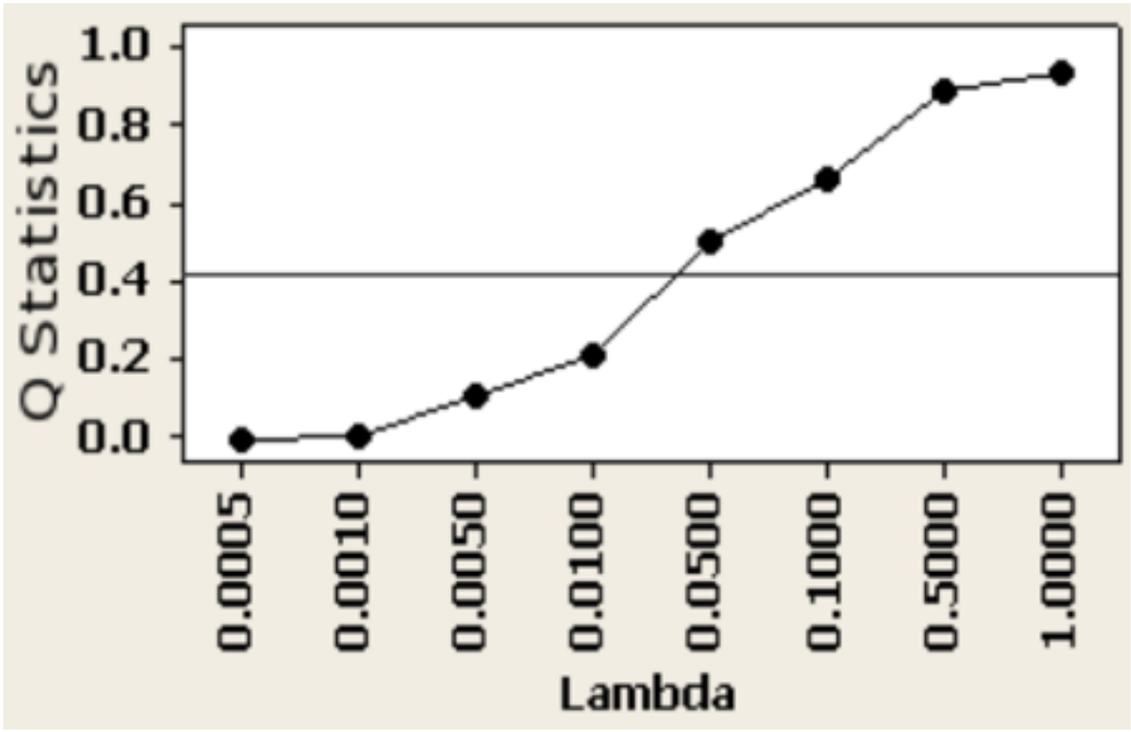
Three different drifting times:
1 time step
0.25N
0.5N

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.

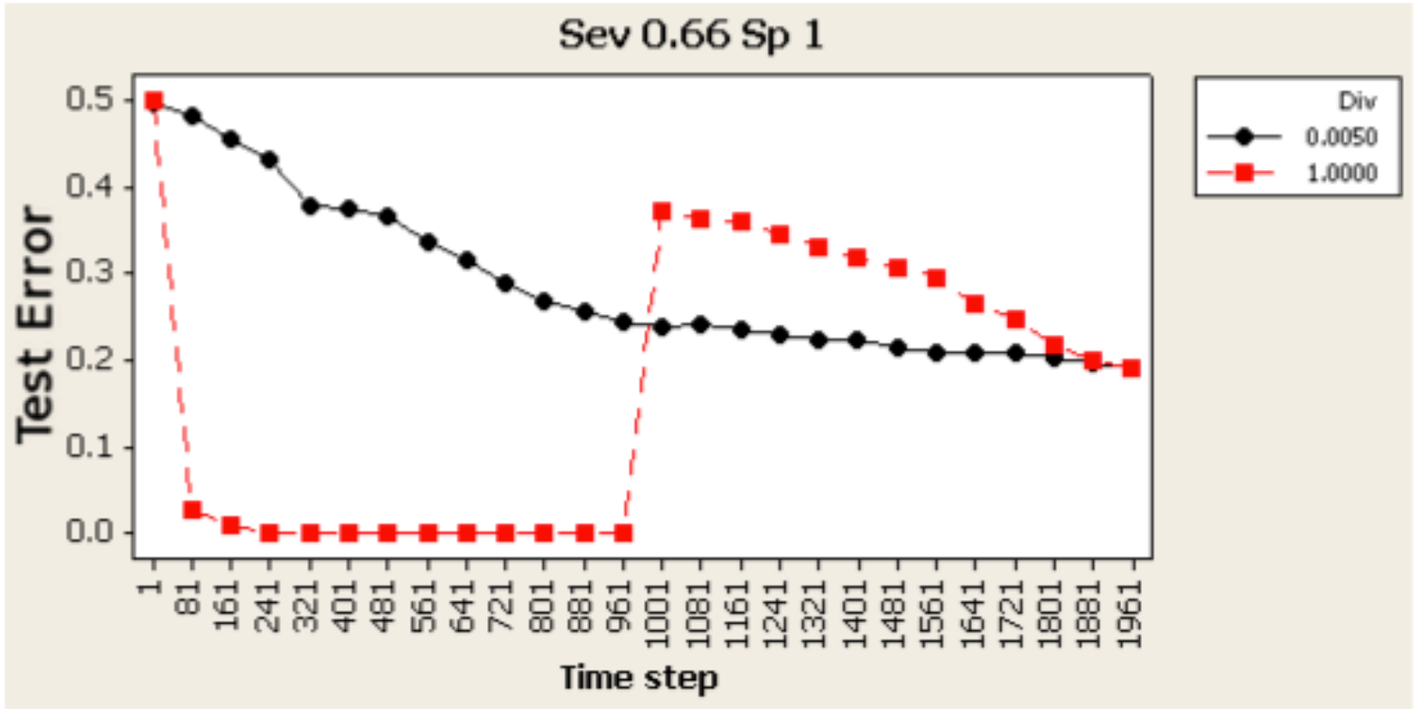
Data sets: same as in preliminary study.

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.

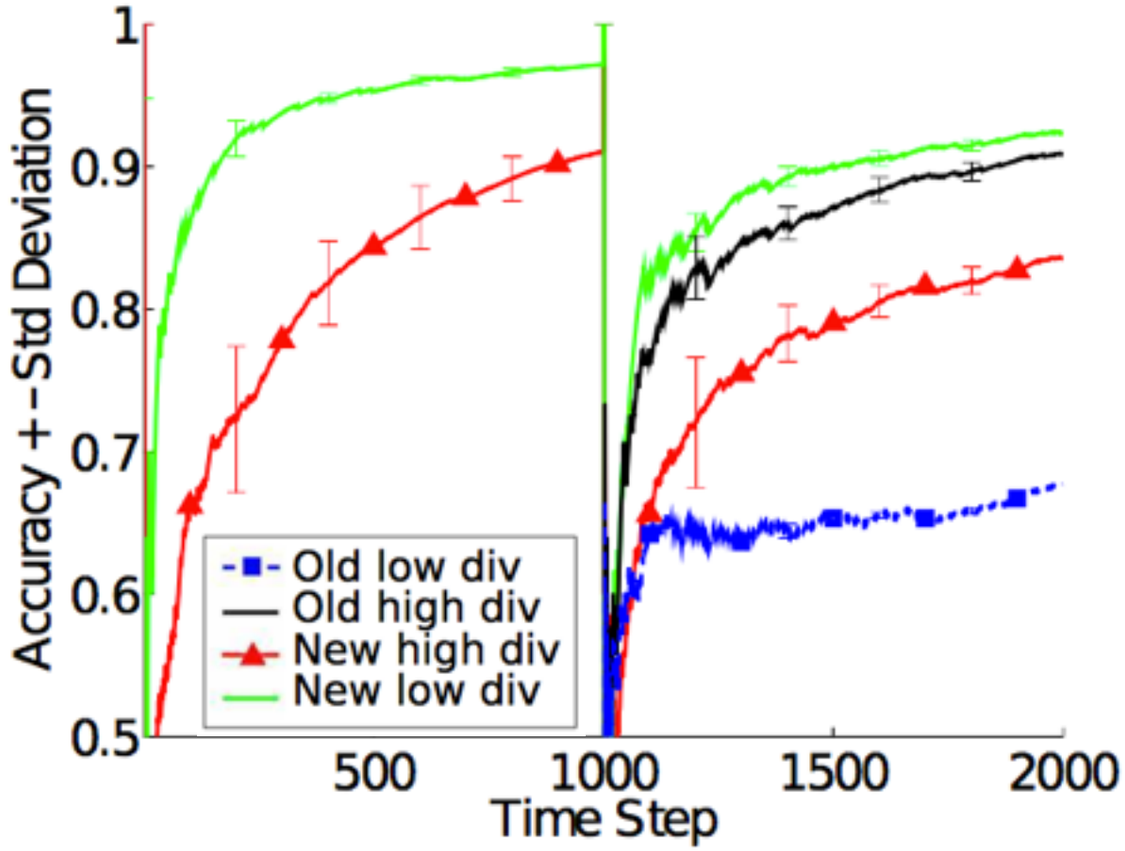
Base learners: 25 DTs

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).

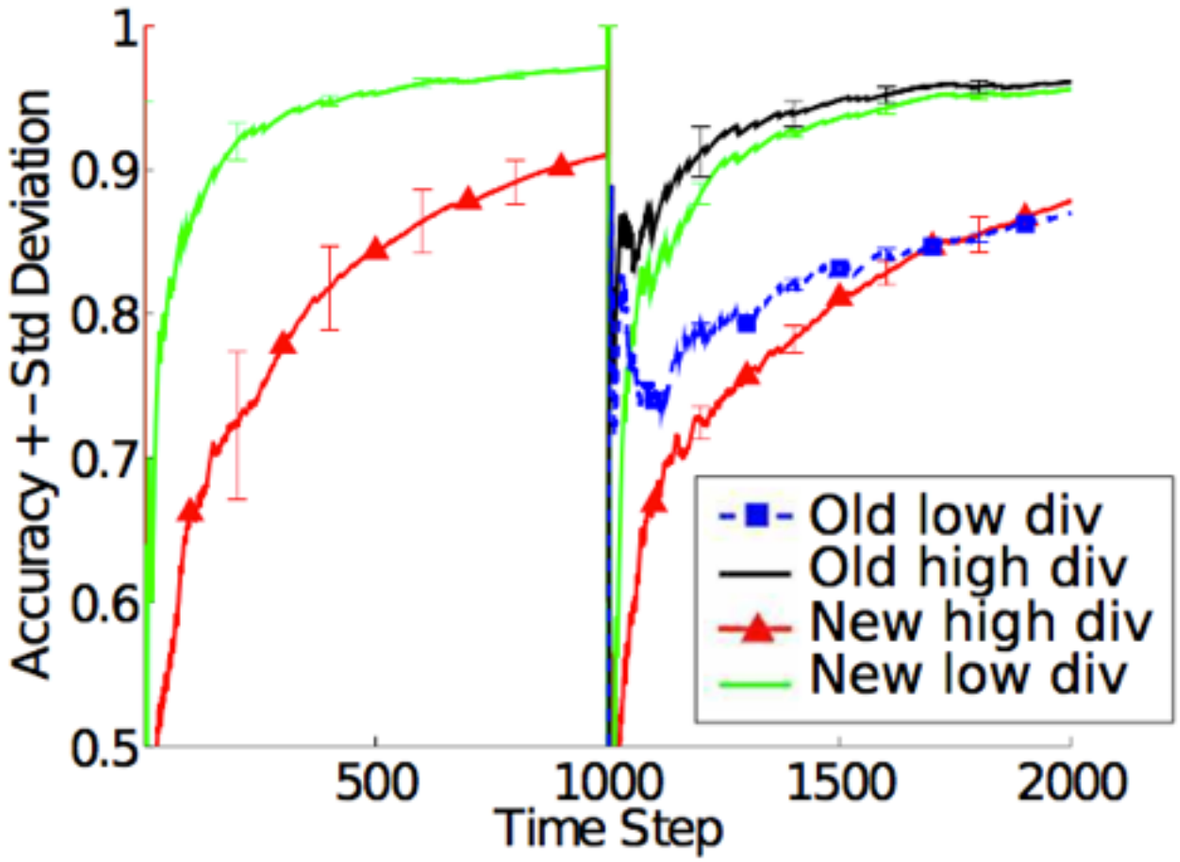


* Sample result for Circle Problem

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!

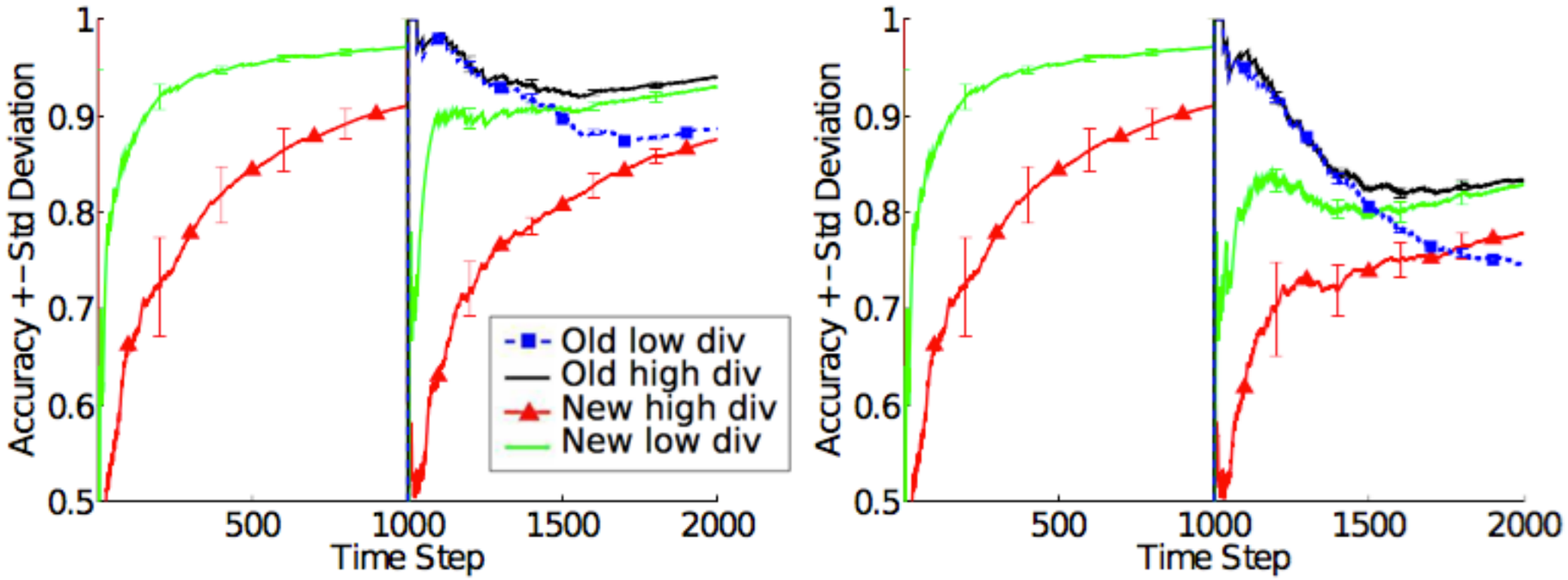


* Sample result for Circle Problem

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

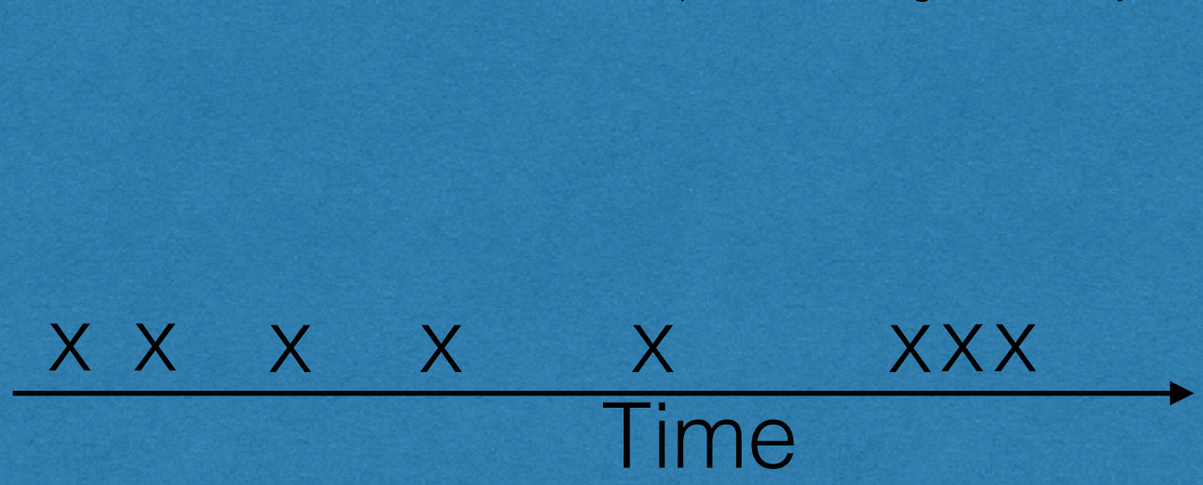
Diversity for Dealing with Drifts (DDD)

Before change detection

Change Detection Method

Early Drift Detection Method

BAENA-GARCIA, M., DEL CAMPO-AVILA, J., FIDALGO, R. and BIFET, A. “Early drift detection method”, Proc. of the 4th ECML PKDD International Workshop on Knowledge Discovery From Data Streams, Berlin, 2006, pp. 77–86.



p' = avg distance between errors

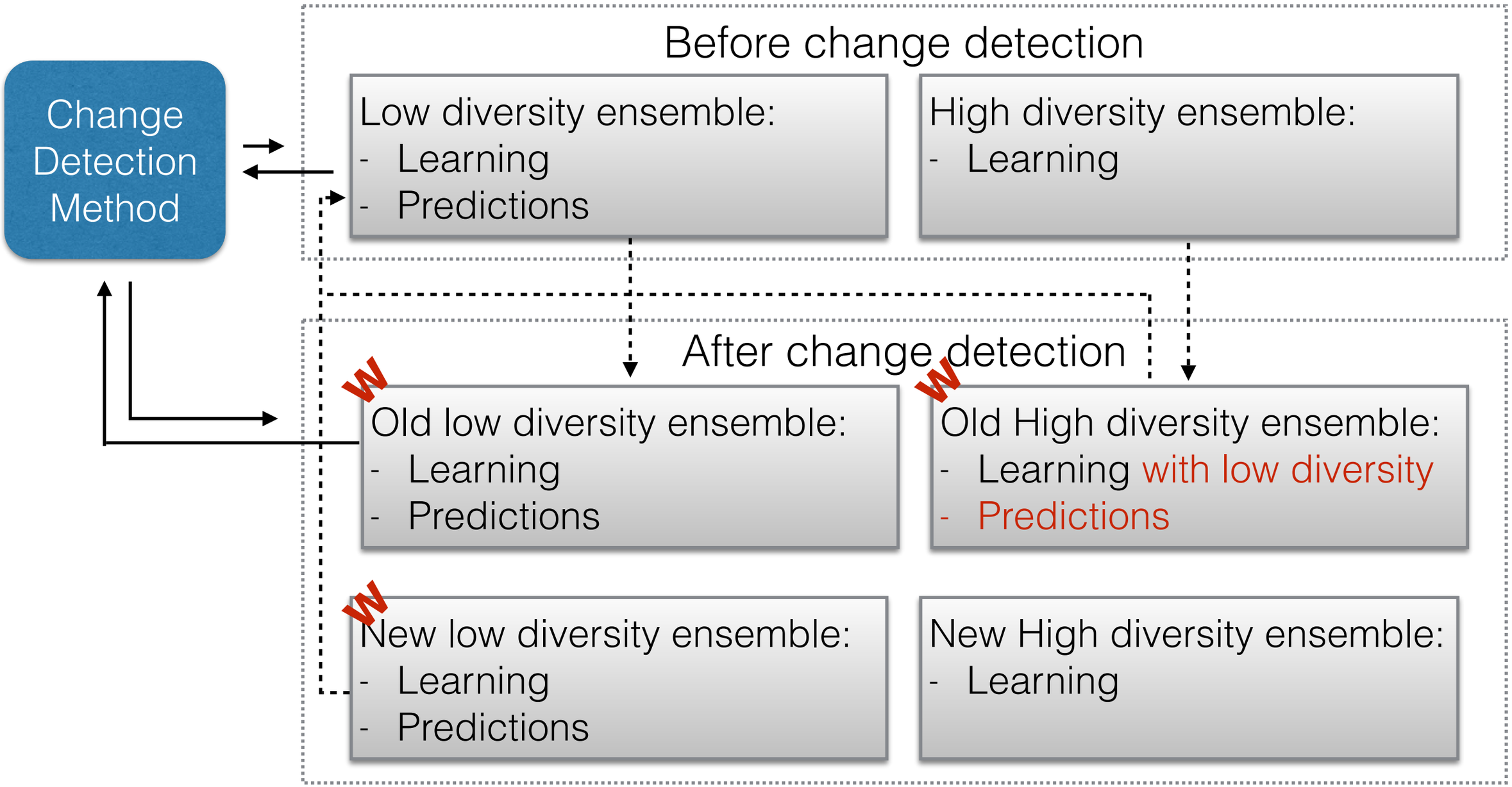
s' = standard deviation

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

Predictions

L. Minku and X. Yao. "DDD: A New Ensemble Approach For Dealing With Concept Drift.", IEEE Transactions on Knowledge and Data Engineering, 24(4):619-633, 2012.

Diversity for Dealing with Drifts (DDD)



L. Minku and X. Yao. "DDD: A New Ensemble Approach For Dealing With Concept Drift.", IEEE Transactions on Knowledge and Data Engineering, 24(4):619-633, 2012.

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).

  M. Baena-Garcıa, J. Del Campo-Avila, R. Fidalgo, and A. Bifet, “Early drift detection method,” in Proc. of the 4th ECML PKDD International Workshop on Knowledge Discovery From Data Streams, Berlin, 2006, pp. 77–86.
- DDD vs Dynamic Weighted Majority (DWM).

  KOLTER, J. and MALOOF, M., “Dynamic weighted majority: An ensemble method for drifting concepts,” Journal of Machine Learning Research, 8:2755–2790, 2007.
- 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.

Probl.	Equation	Fixed Values	Before→After Drift	Sev.
Circle	$(x-a)^2+(y-b)^2 \leq r^2$	$a=.5$ $b=.5$	$r=.2 \rightarrow .3$	16%
			$r=.2 \rightarrow .4$	38%
			$r=.2 \rightarrow .5$	66%
SineV	$y \leq a\,sin(bx+c)+d$	$a=1$ $b=1$ $c=0$	$d=-2 \rightarrow 1$	15%
			$d=-5 \rightarrow 4$	45%
			$d=-8 \rightarrow 7$	75%
SineH	$y \leq a\,sin(bx+c)+d$	$a=5$ $d=5$ $b=1$	$c=0 \rightarrow -\pi/4$	36%
			$c=0 \rightarrow -\pi/2$	57%
			$c=0 \rightarrow -\pi$	80%
Line	$y \leq -a_0+a_1x_1$	$a_1=.1$	$a_0=-.4 \rightarrow -.55$	15%
			$a_0=-.25 \rightarrow -.7$	45%
			$a_0=-.1 \rightarrow -.8$	70%
Plane	$y \leq -a_0+a_1x_1+a_2x_2$	$a_1=.1$ $a_2=.1$	$a_0=-2 \rightarrow -2.7$	14%
			$a_0=-1 \rightarrow -3.2$	44%
			$a_0=-.7 \rightarrow -4.4$	74%
Bool	$(color\ eq_1\ a\ op_1\ shape\ eq_2\ b)\ op_2\ size\ eq_3\ c$	$c=S \vee M \vee L$ $op_2 \wedge$ $eq_{1,2,3}=$	$a=R, op_1 \wedge b=R \rightarrow R \vee T$	11%
			$a=R, b=R, op_1 \wedge \rightarrow \vee$	44%
			$a=R \rightarrow R \vee G, b=R \rightarrow R \vee T, op_1 \wedge \rightarrow \vee$	67%

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.

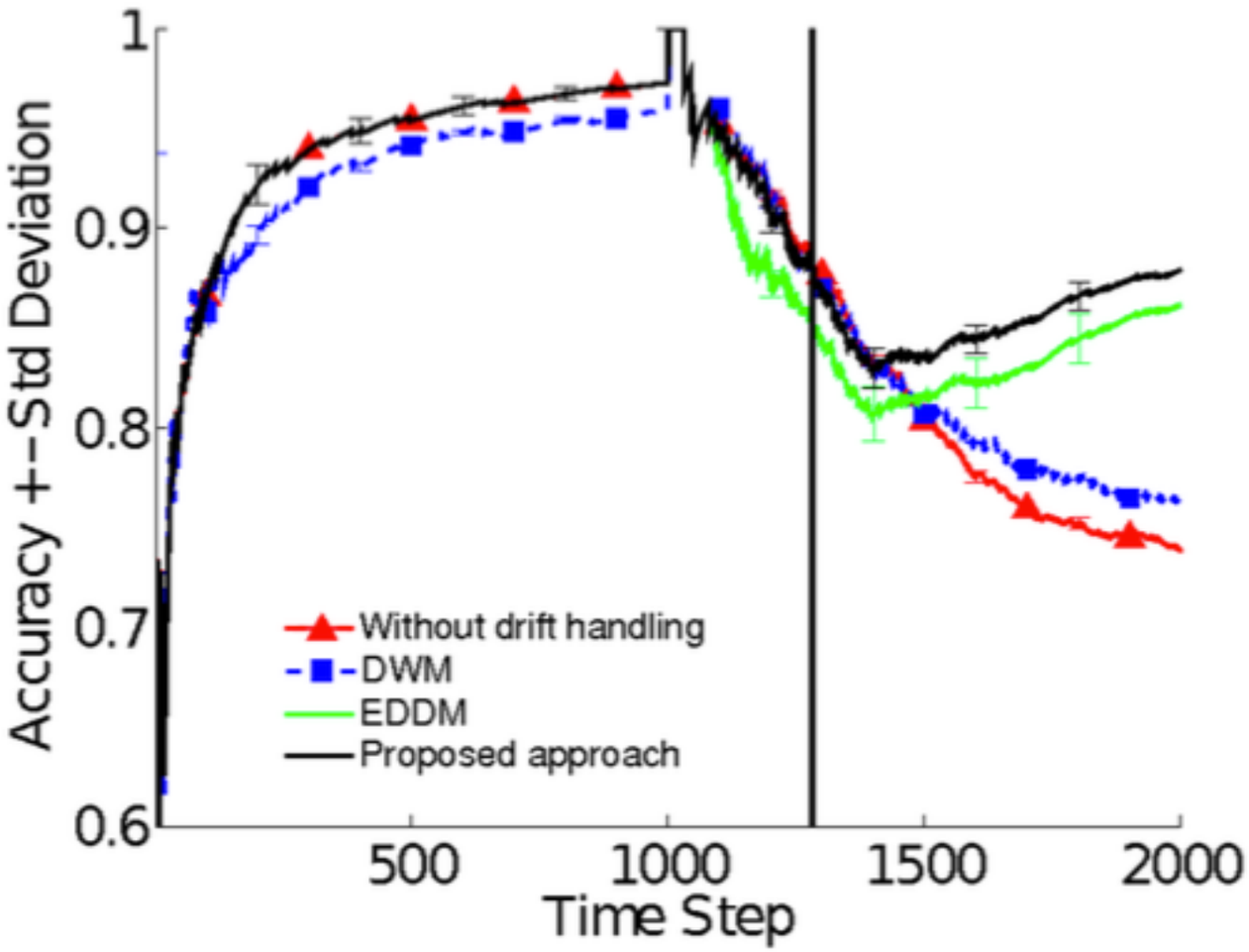
Electricity price prediction:

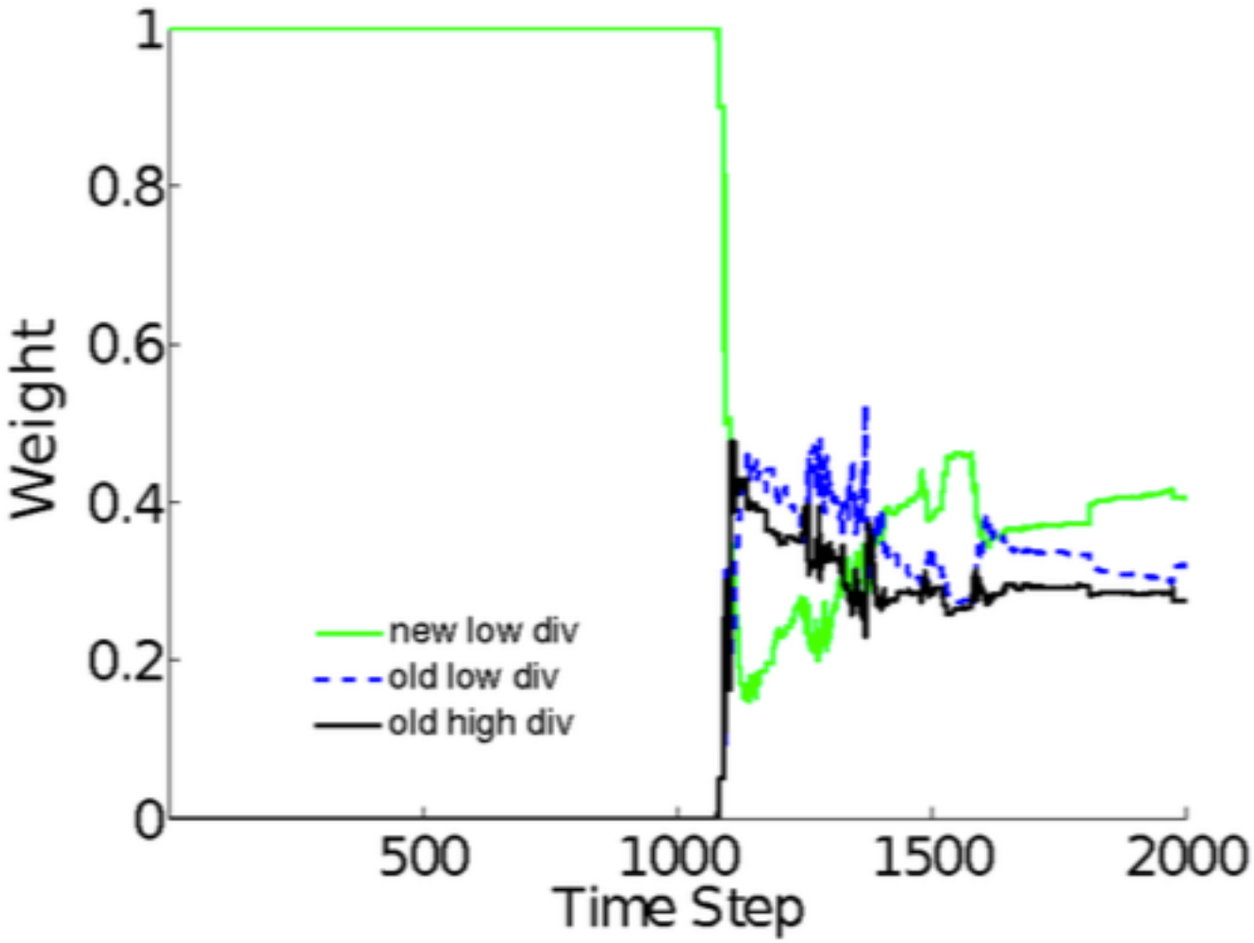
- 45,312 examples
- 4 input attributes (time stamp, day of the week and 2 electricity demand values)
- 1 target class: increase / decrease in price.

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

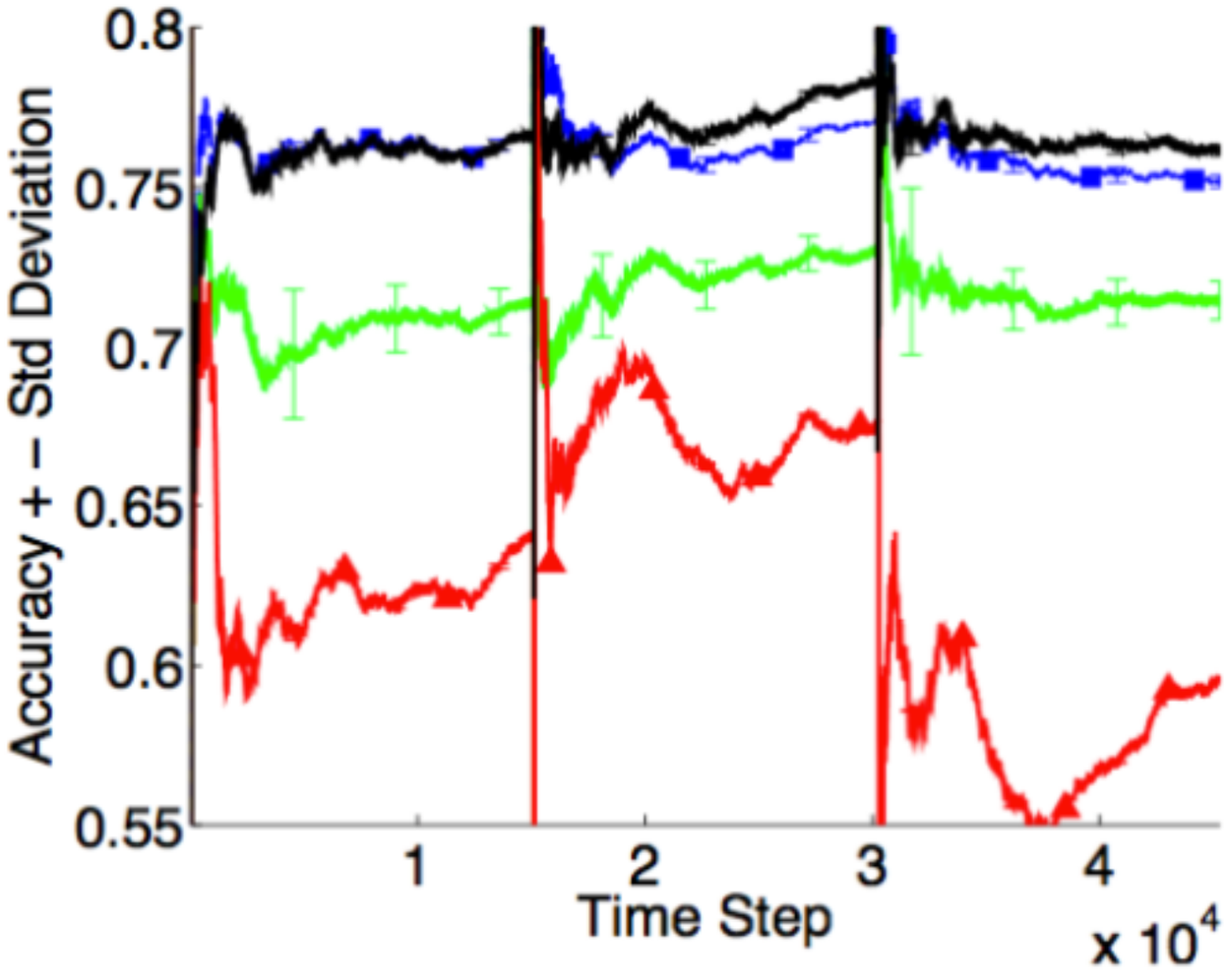


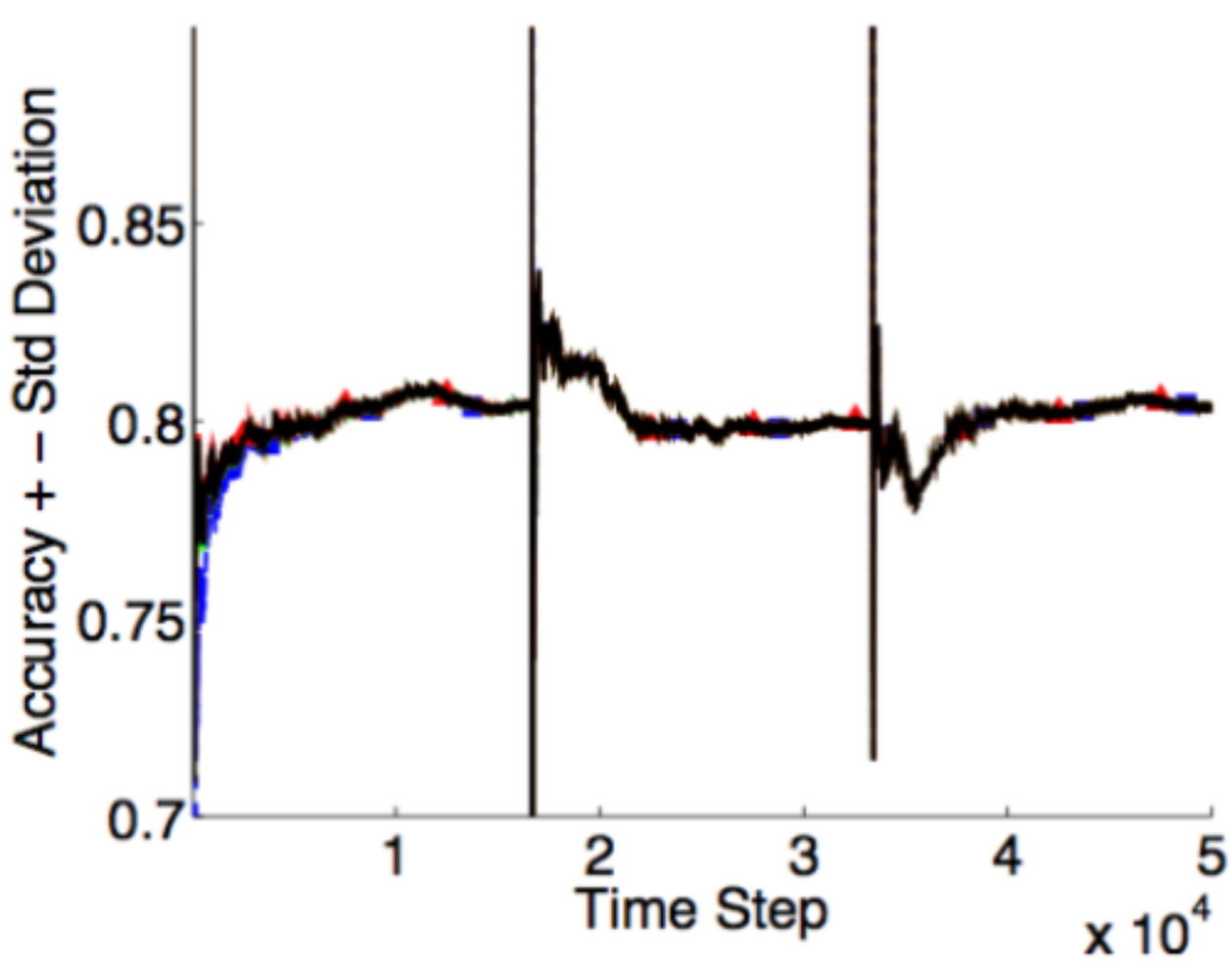


Circle – high severity and low speed.

- 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





Electricity (NB) and Credit Card (MLP).

Type of Drifts for Which DDD Behaves Better

T-tests with Holm-Bonferroni corrections after drifts:

	Win	Tie	Loss
DDD vs EDDM	45%	48%	7%
DDD vs DWM	59%	25%	15%

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

Picture: http://www.boeing.com/defense-space/space/hsfe_shuttle/index.html

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:

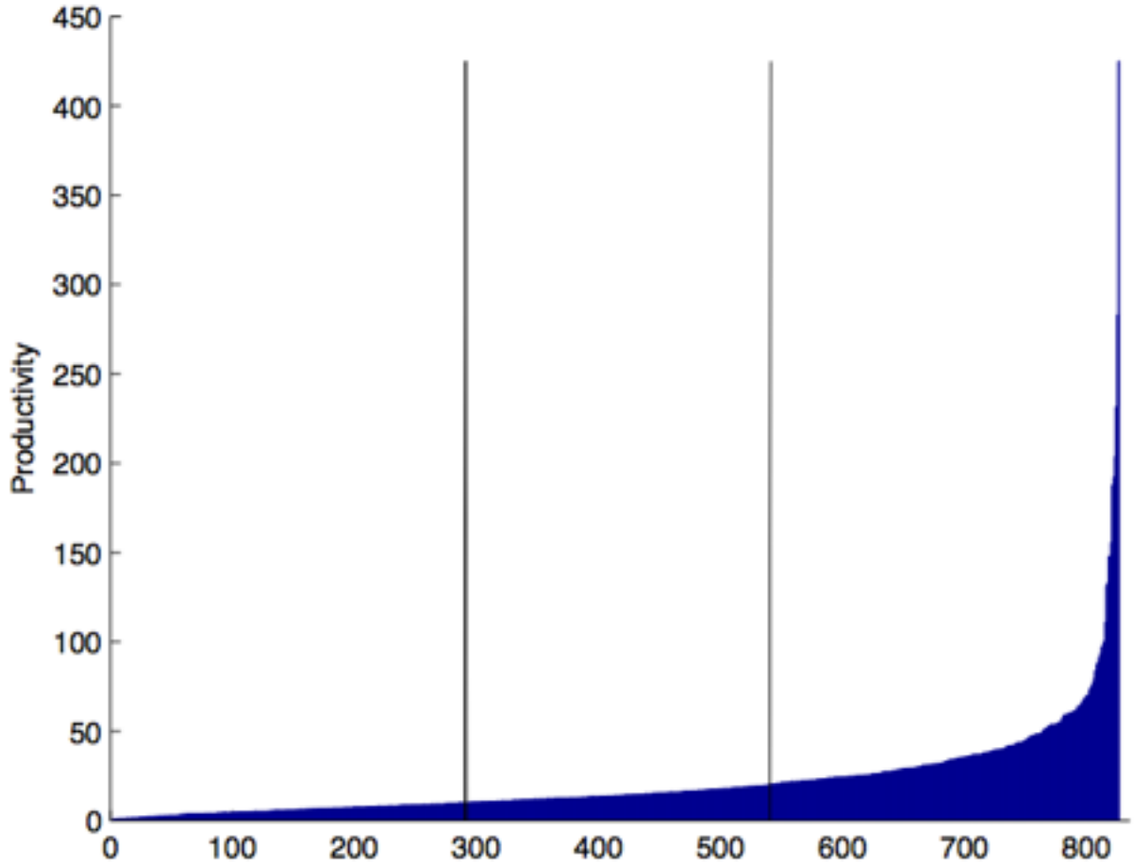
- Regression trees.

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

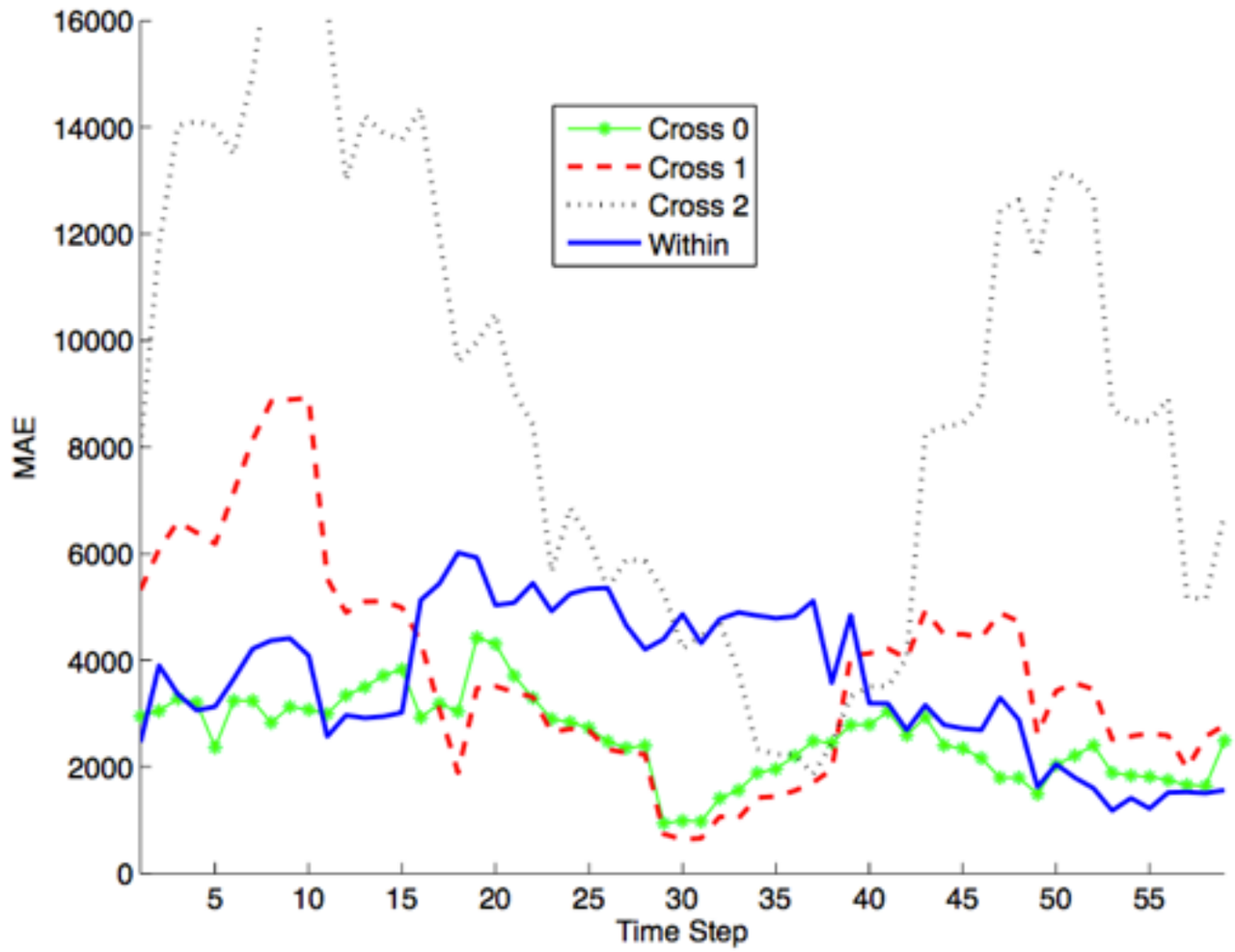
- CocNasaCoc81: 60 WC Nasa projects 1980s–1990s. 63 CC projects up to 1980.
- CocNasaCoc81Nasa93: additional 93 Nasa projects 1970s–1980s, considered CC.

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

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

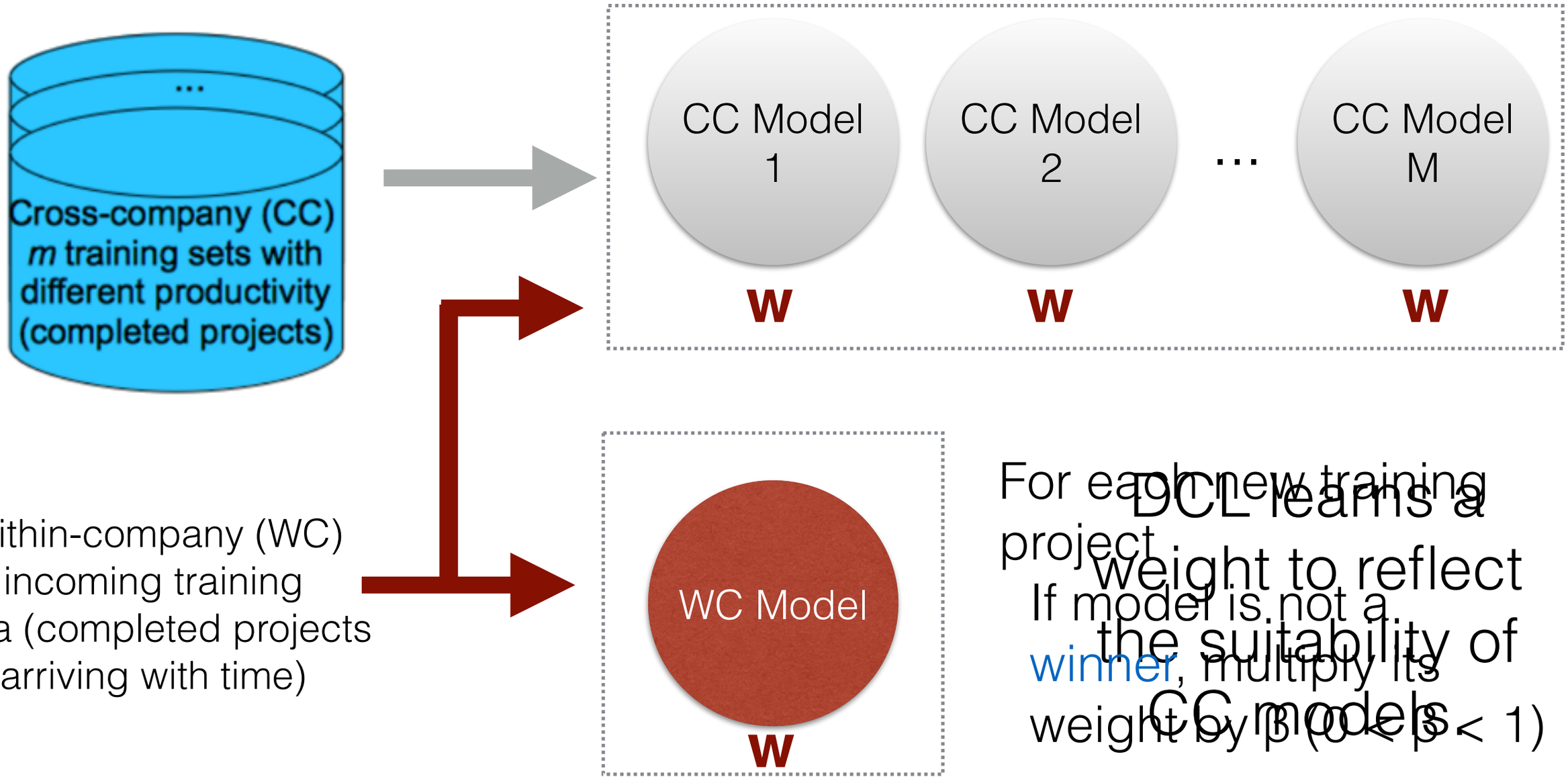
Analysis

- 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)



MINKU, L.; YAO, X. . "Can Cross-company Data Improve Performance in Software Effort Estimation?", Proceedings of the 8th International Conference on Predictive Models in Software Engineering (PROMISE'2012), p. 69-78, Lund, Sweden, September 2012, doi: 10.1145/2365324.2365334.

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) = $\sum_1^n \frac{|y_i - \hat{y}_i|}{n}$
- Standardised accuracy = (1 - MAE / MAErguess) * 100
- Effect size = |MAEc - MAE| / MAEc

Base learners:

- Regression trees.

Data sets:

- Same as in previous study.

Overall Performance

Approach	Rank Avg	Rank Std	P-value
DCL	1	[illegible]	[illegible]
CC-DW			
WC-DW			
RT			

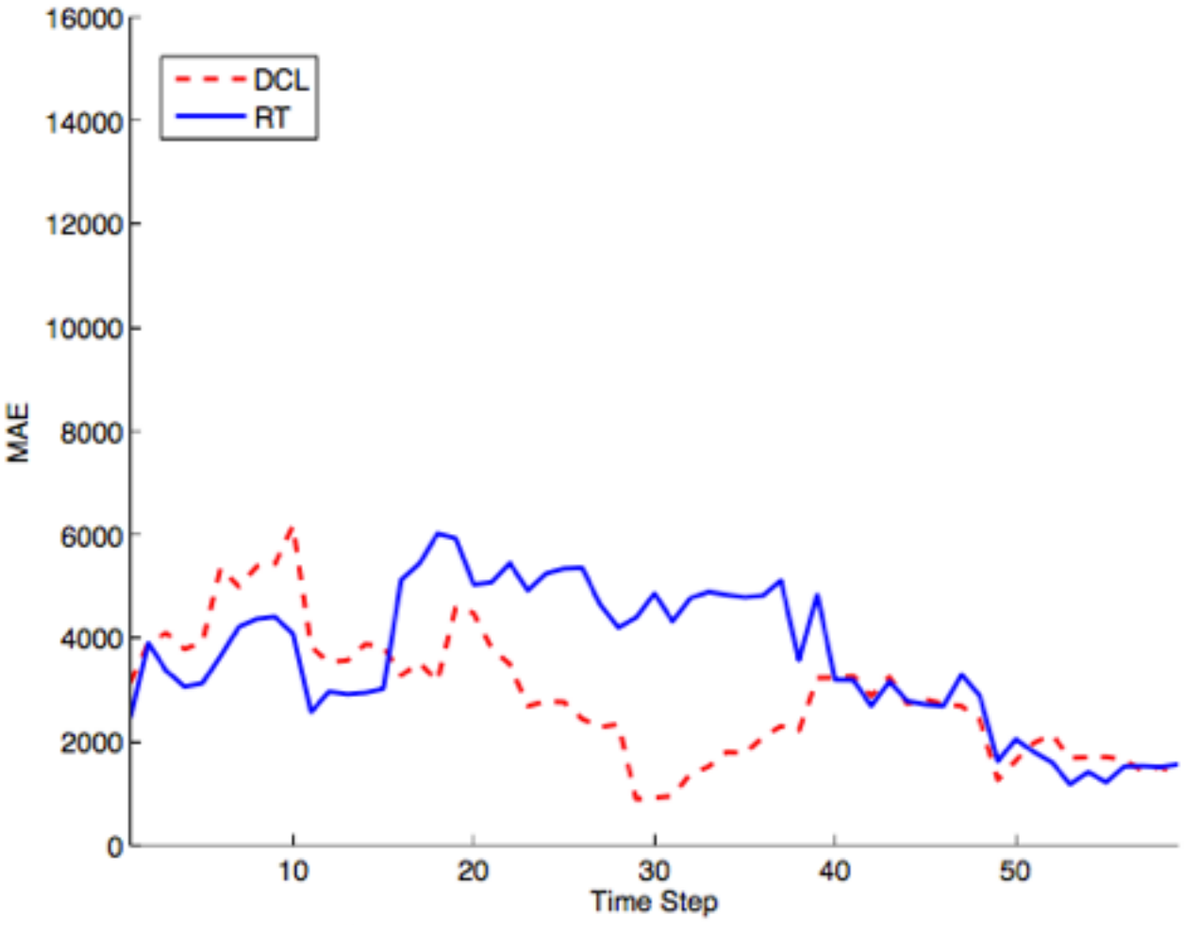
Effect size against RT

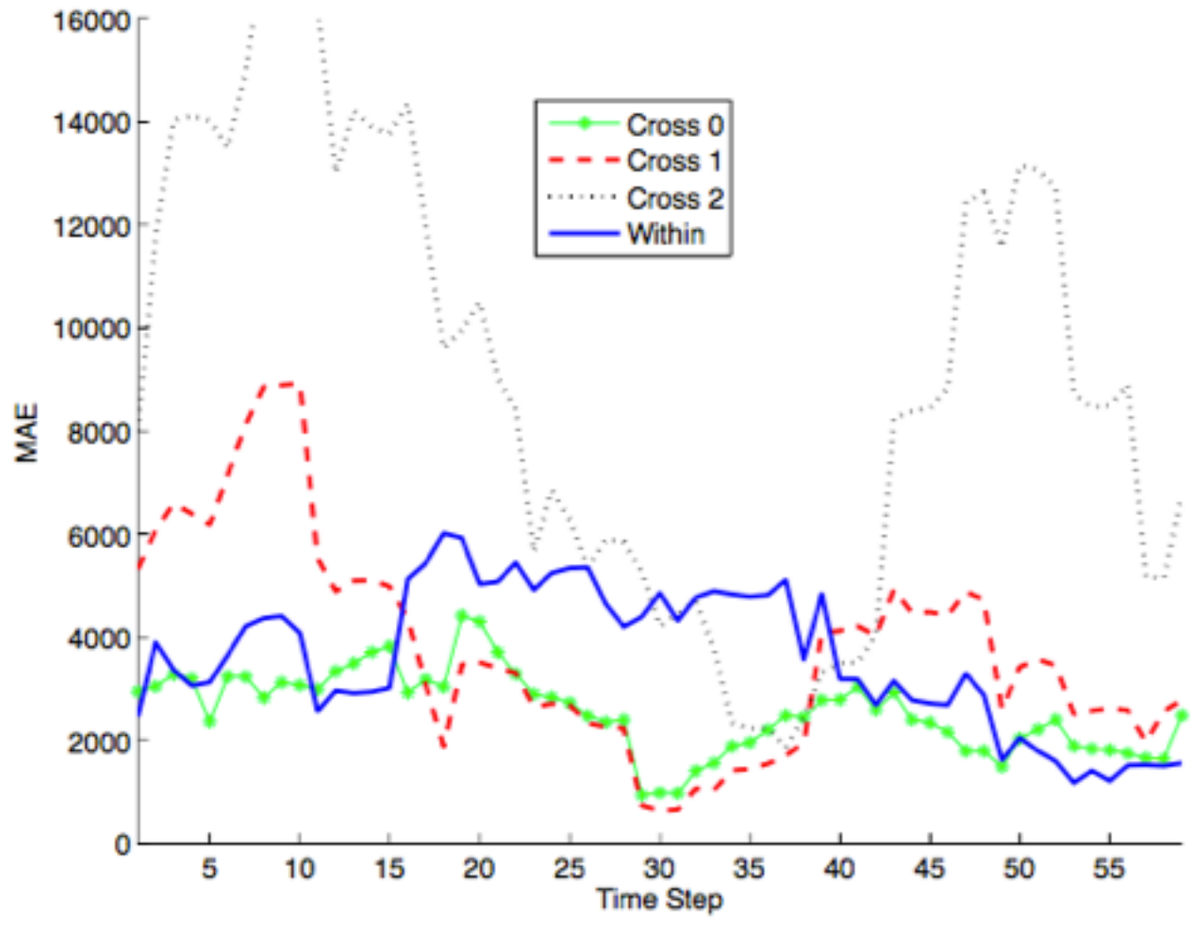
Data Set	WC-DWM	CC-DWM	DCL
ISBSG2000	0.0868	0.1490	0.3187
ISBSG2001	0.2501	0.5028	0.5474
ISBSG	0.0376	0.0876	0.1811
CocNasaCoc81	0.0768	0.2978	0.4541
CocNasaCoc81Nasa93	0.0768	0.6014	0.8378

- Friedman (MAE across data sets): overall MAE of the approaches
- ct
- m

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

Performance Throughout Time



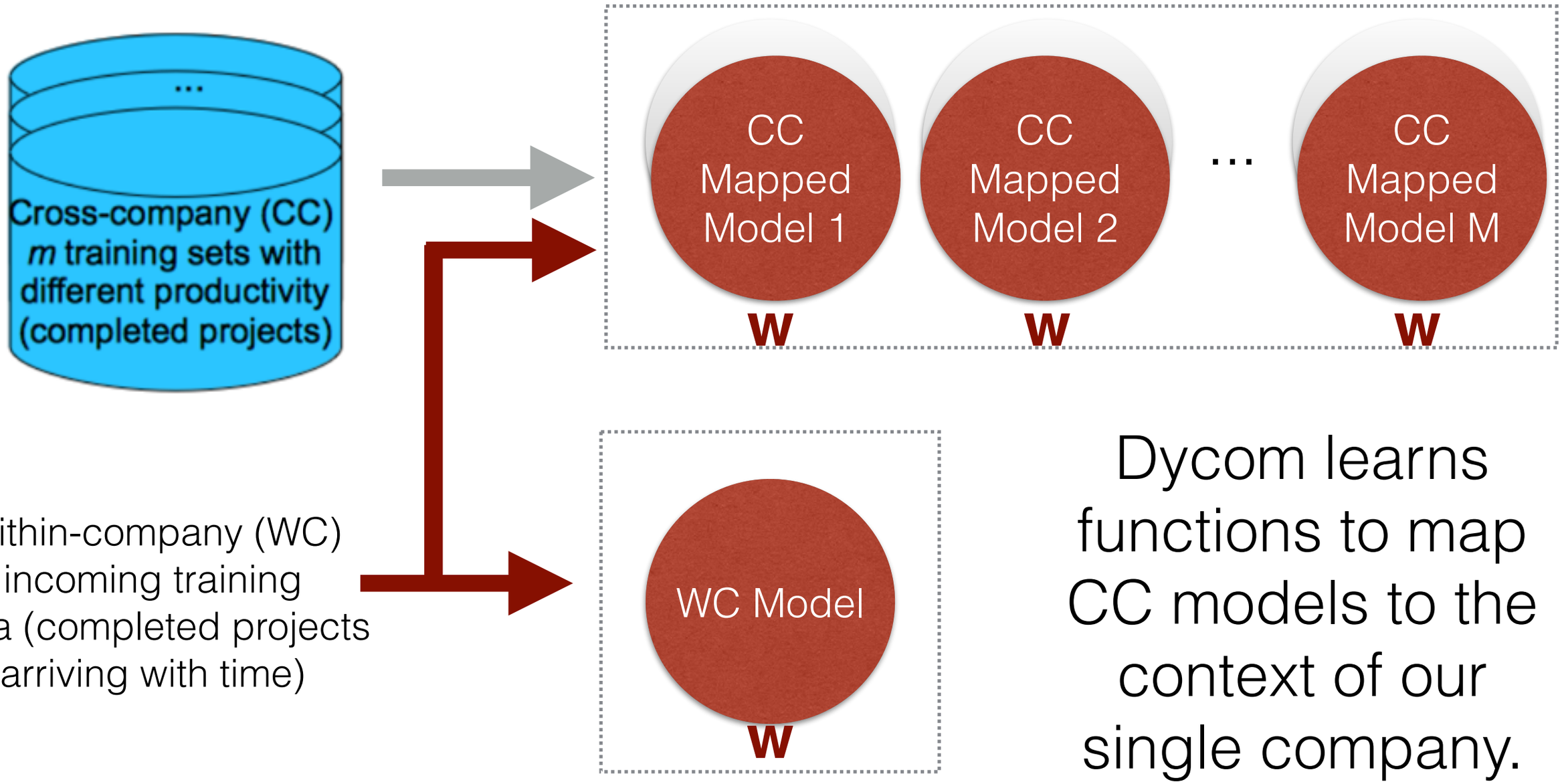


ISBSG'2001 (≈ 1.5 year)

- Considerable improvement over periods of several months. Improvements
- improvement.

DCL uses CC models only when they are directly useful.

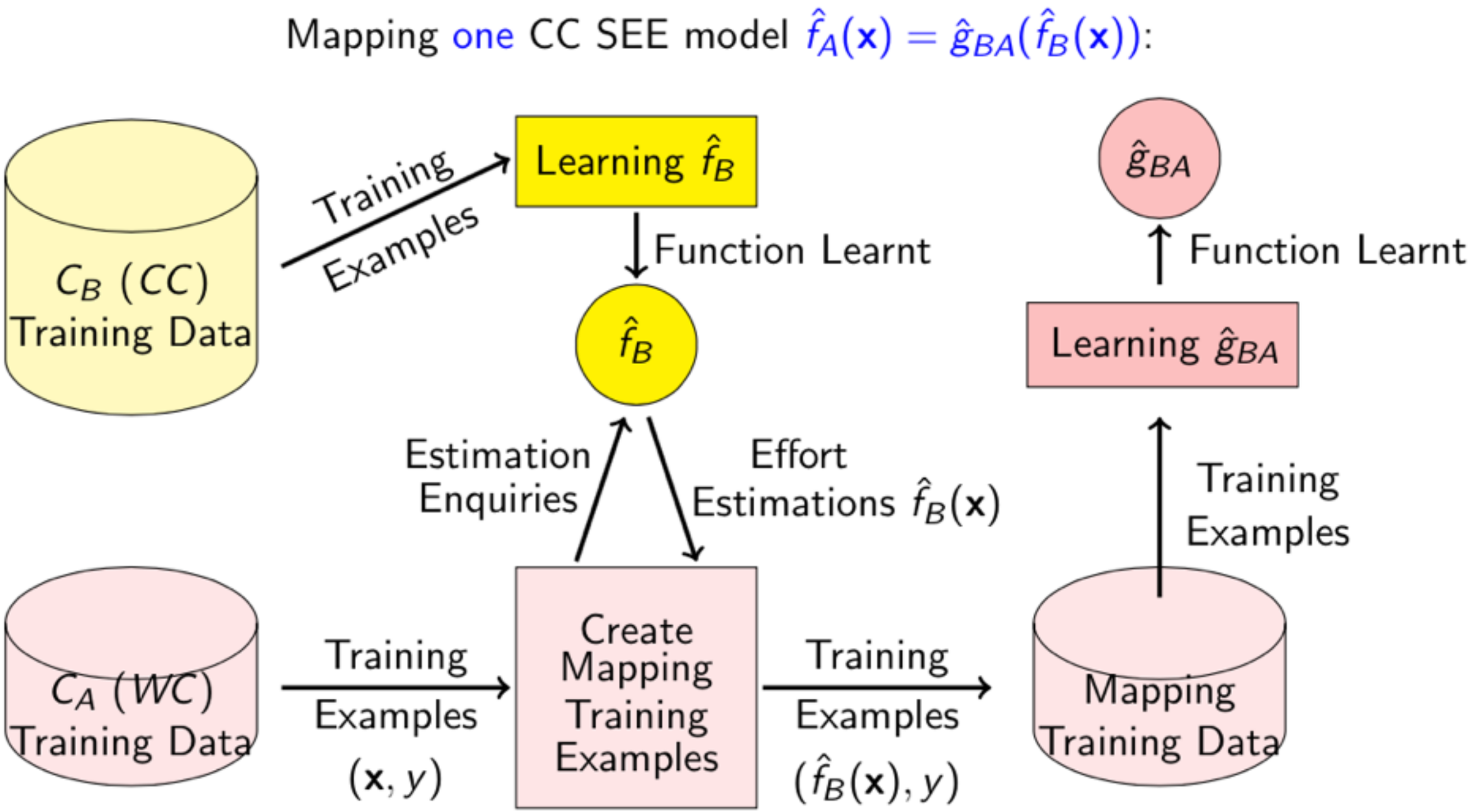
Dynamic Cross-Company Mapped Model Learning (Dycom)



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

MINKU, L. L.; YAO, X.; "How to Make Best Use of Cross-company Data in Software Effort Estimation?", Proceedings of the 36th International Conference on Software Engineering (ICSE'14), p. 446-456, May 2014, doi: 10.1145/2568225.2568228.

Mapping Training Examples



Dycom's Mapping Function

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

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

where ***lr*** is a smoothing factor that allows tuning the emphasis on more recent examples.

Experimental Study

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.

KitchenMax: 145 WC Kitchenham projects, 62 CC projects. Input attributes: functional size.

- Target: effort in person-hours.

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

Performance Measures

$MAE = \frac{1}{T}\sum_{i=1}^{T}|\hat{y}_i - y_i|$;

StdDev = standard deviation of MAE across time steps.

$SA = \left(1 - \frac{MAE}{MAE_{rguess}}\right) \cdot 100,$

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

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

where $\bar{\hat{y}}$ and $\bar{y}$ are the average predicted and average actual efforts, respectively;

$LSD = \sqrt{\frac{\sum_{i=1}^{T}\left(e_i + \frac{s^2}{2}\right)^2}{T-1}}$, where s^2 is an estimator of the variance of the residual e_i and $e_i = \ln y_i - \ln \hat{y}_i$;

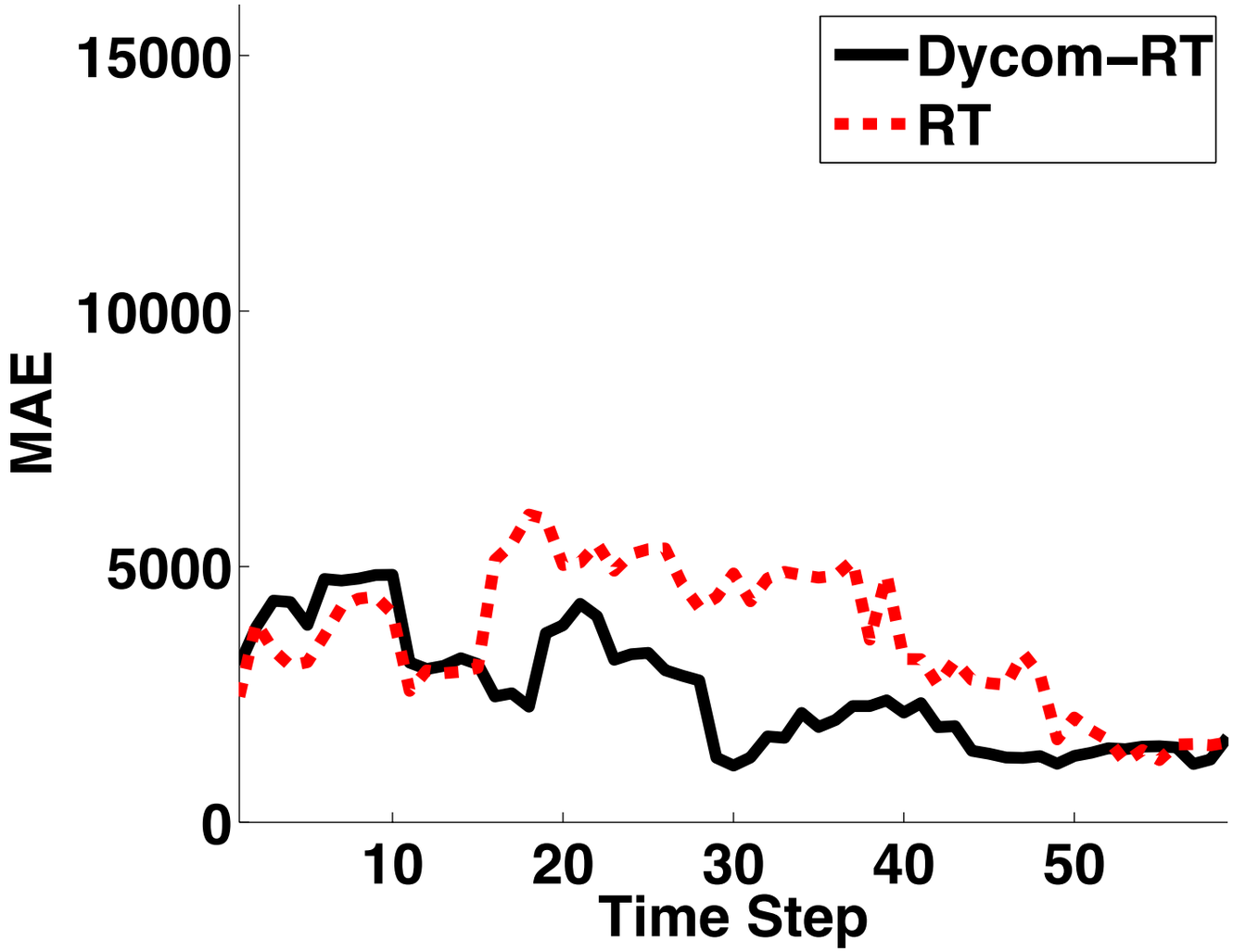
Calculated over next 10 examples.

Overall Performance

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

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

Performance Throughout Time - Sample Result



* Sample result for ISBSG2001 data set.

Dycom can achieve similar / better performance while using only 10% of WC data.

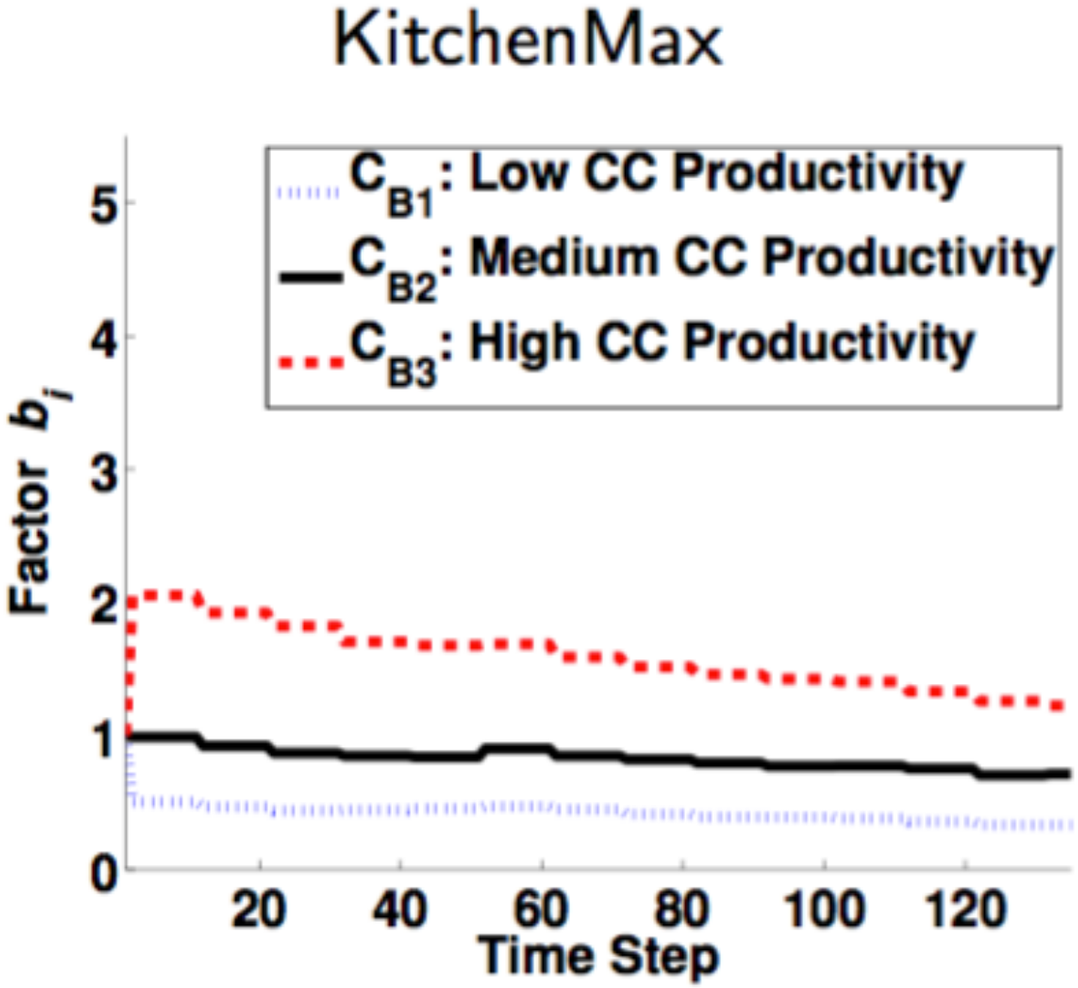
Summary of Dycom's Results

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.

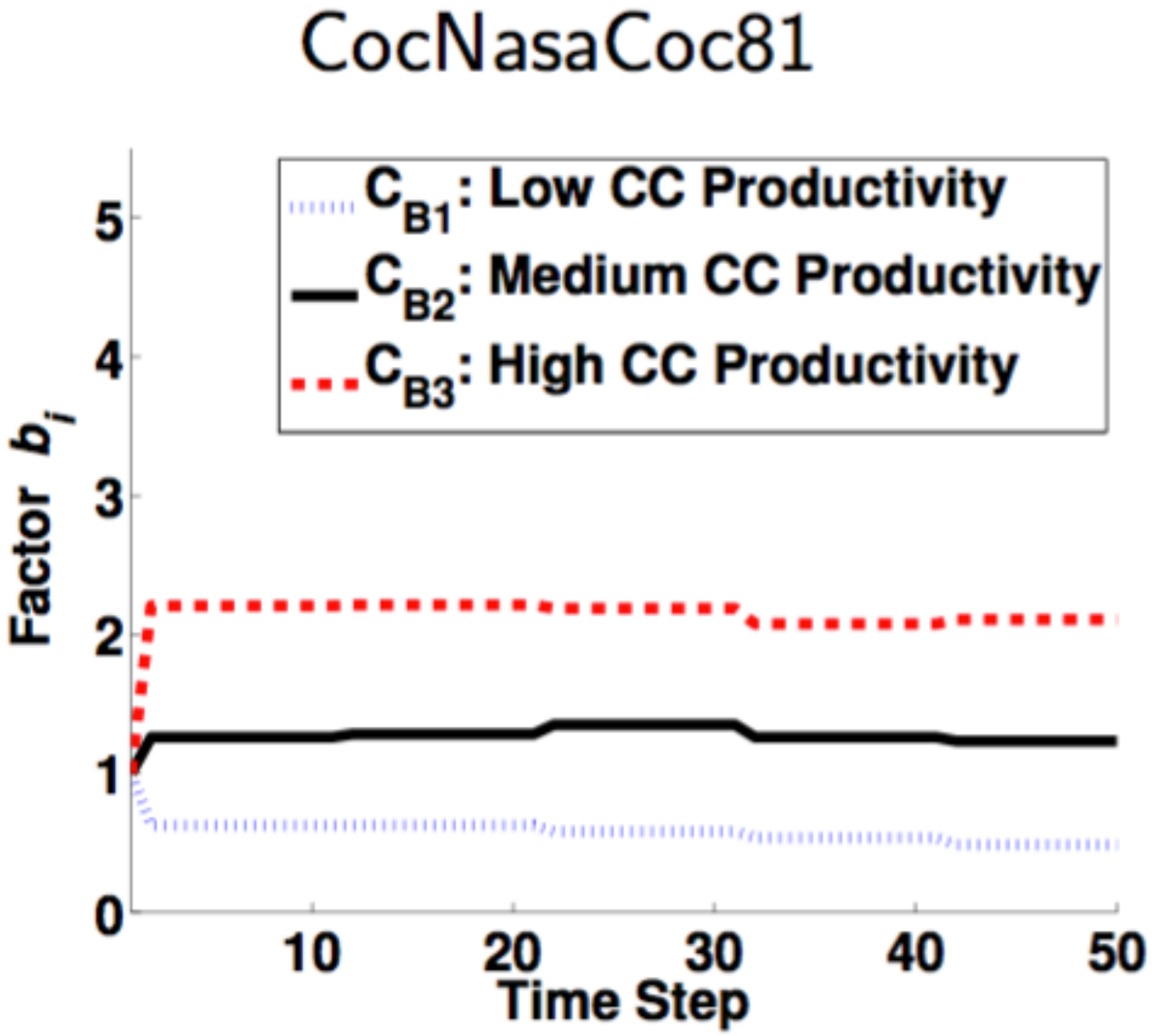
Dycom Insights on Productivity



$$\hat{f}_A(\mathbf{x}) = \hat{f}_{Bi}(\mathbf{x}) \cdot b_i$$

- Initially, our company needs initially 2x effort than company red.
- Later, it needs only 1.2x effort.

Dycom Insights on Productivity



$$\hat{f}_A(\mathbf{x}) = \hat{f}_{Bi}(\mathbf{x}) \cdot b_i$$

- 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:

Feature / Value	Lang. exp		Virtual mach. exp	
	CC	WC	CC	WC
Very low	1	0	1	0
Low	1	0	4	4
Nominal	8	8	8	16
High	10	12	7	0
Very high	0	0	0	0
Extremely high	0	0	0	0

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:

Feature / Value	Lang. exp		Virtual mach. exp	
	CC	WC	CC	WC
Very low	1	0	1	0
Low	1	0	4	4
Nominal	8	8	8	16
High	10	12	7	0
Very high	0	0	0	0
Extremely high	0	0	0	0

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 $p(\boldsymbol{x},w)$:

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

or

Class imbalance changes

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

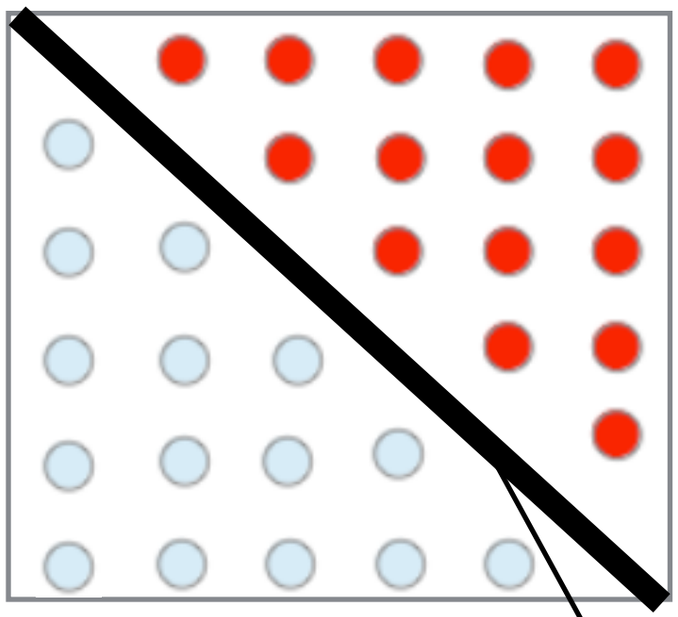
Class-conditional changes

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

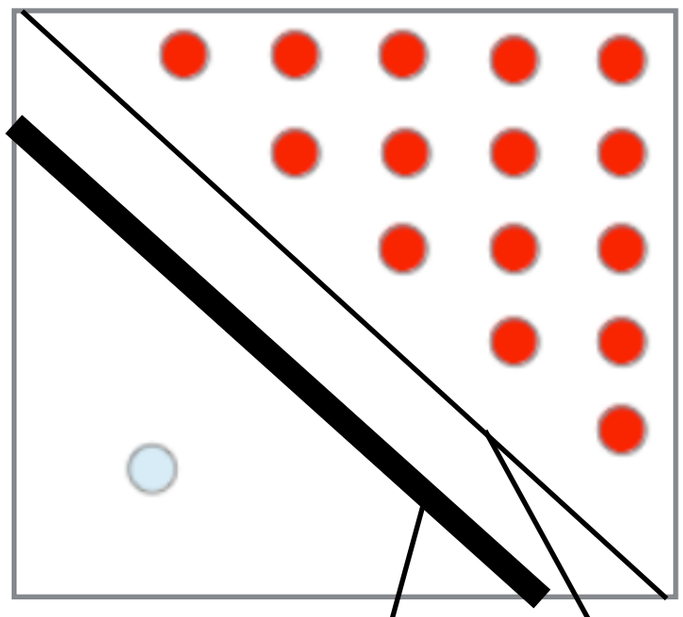
Class Imbalance Vs Class-Conditional Changes

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!



Learnt class boundary = true class boundary



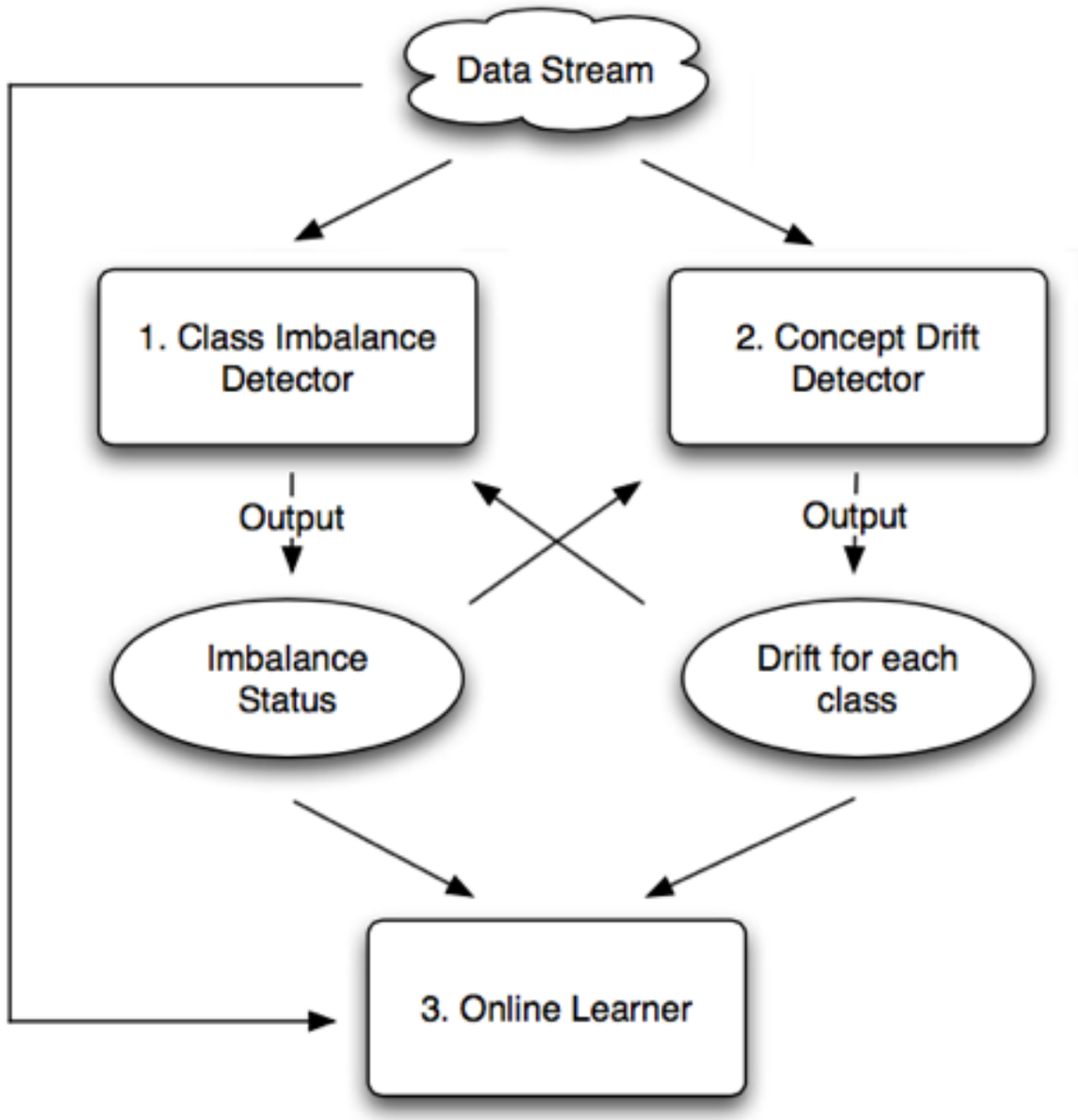
Learnt class boundary

True class boundary

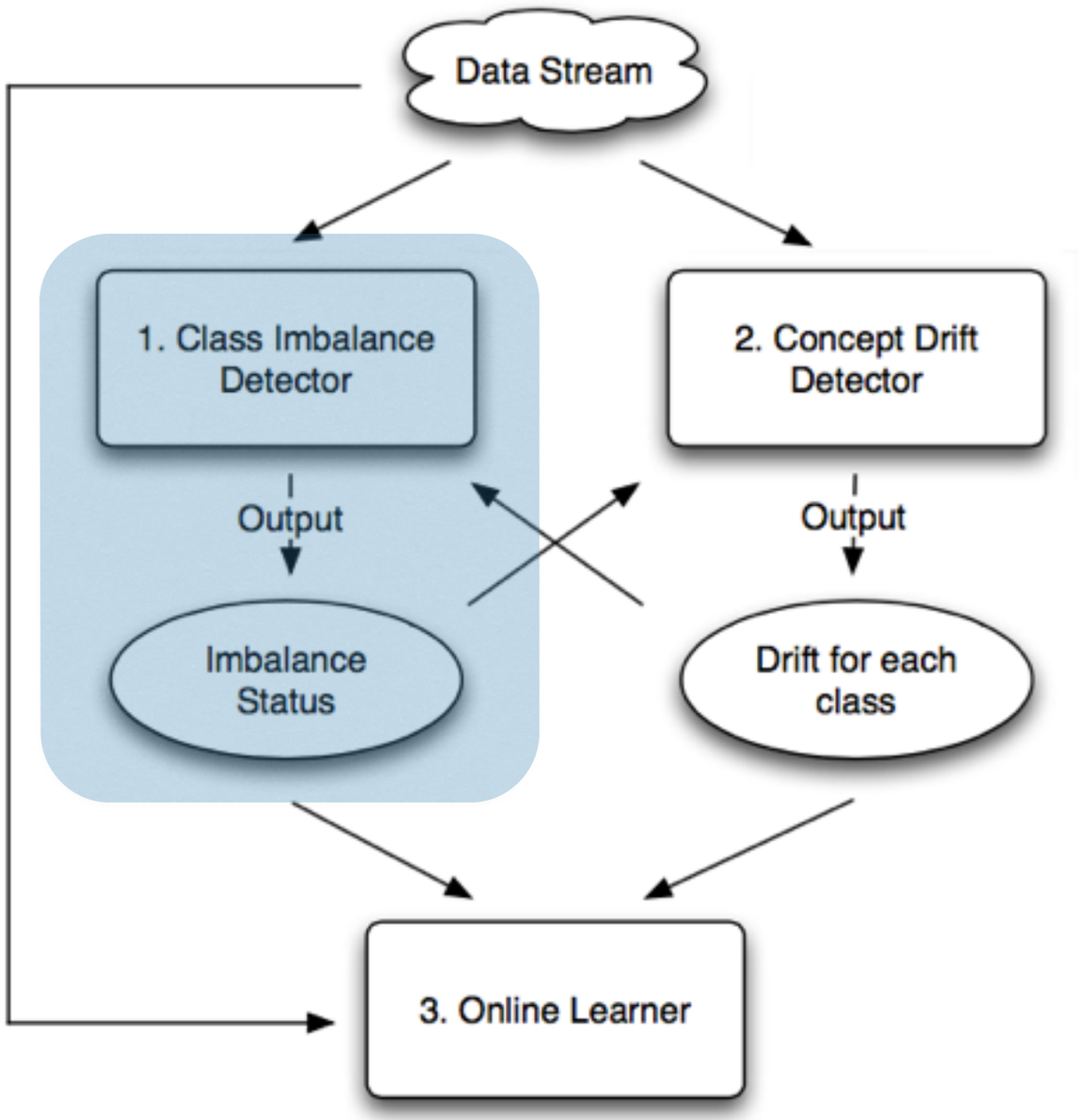
Online Class Imbalance Learning Framework

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

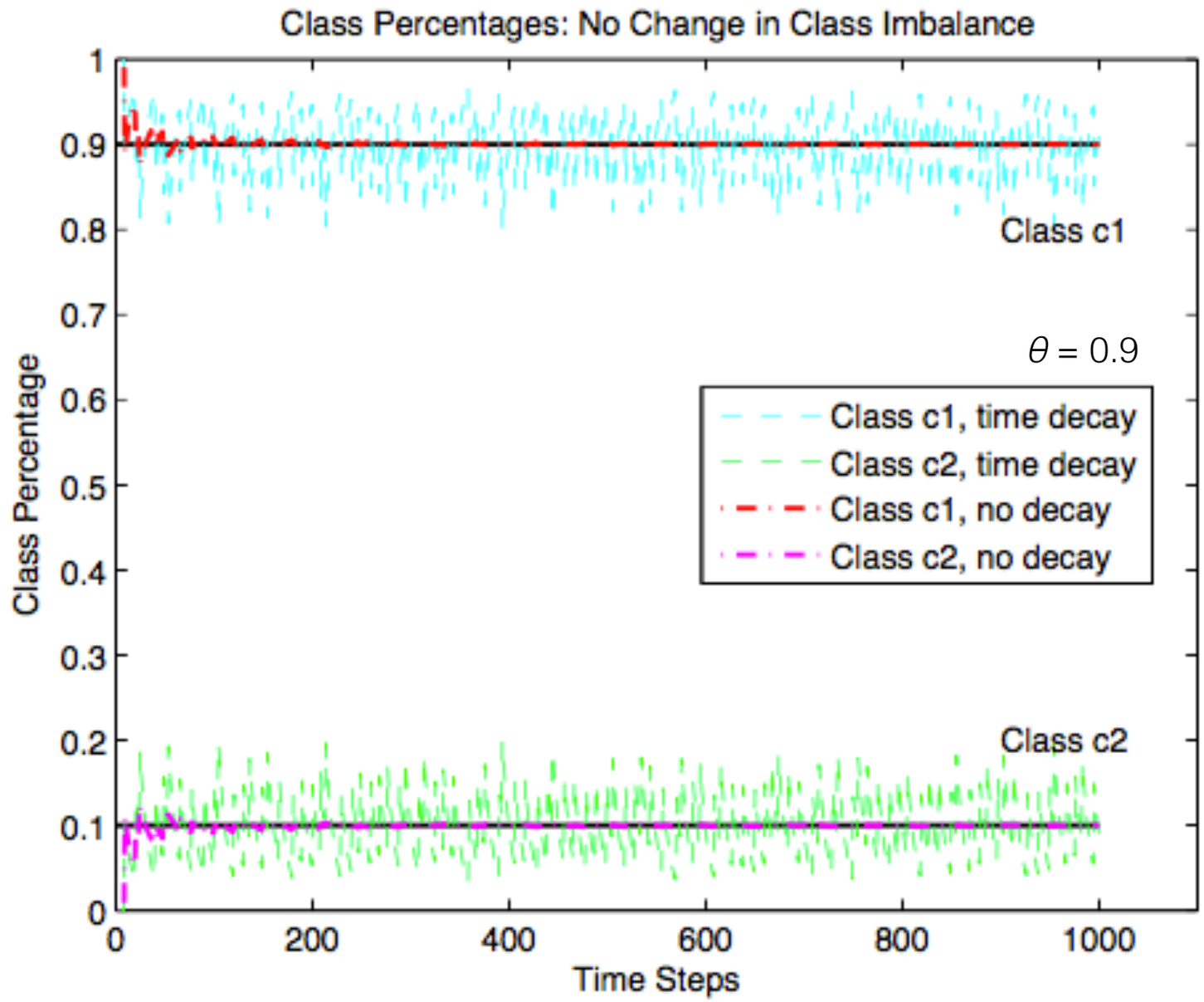
- Suppose an incoming sequence of examples (x_t, c_t).
- Time-decayed class size:

$$w_k^{(t)} = \theta w_k^{(t-1)} + (1-\theta)\left[(x_t, c_k)\right], (k = 1, \ldots, N)$$

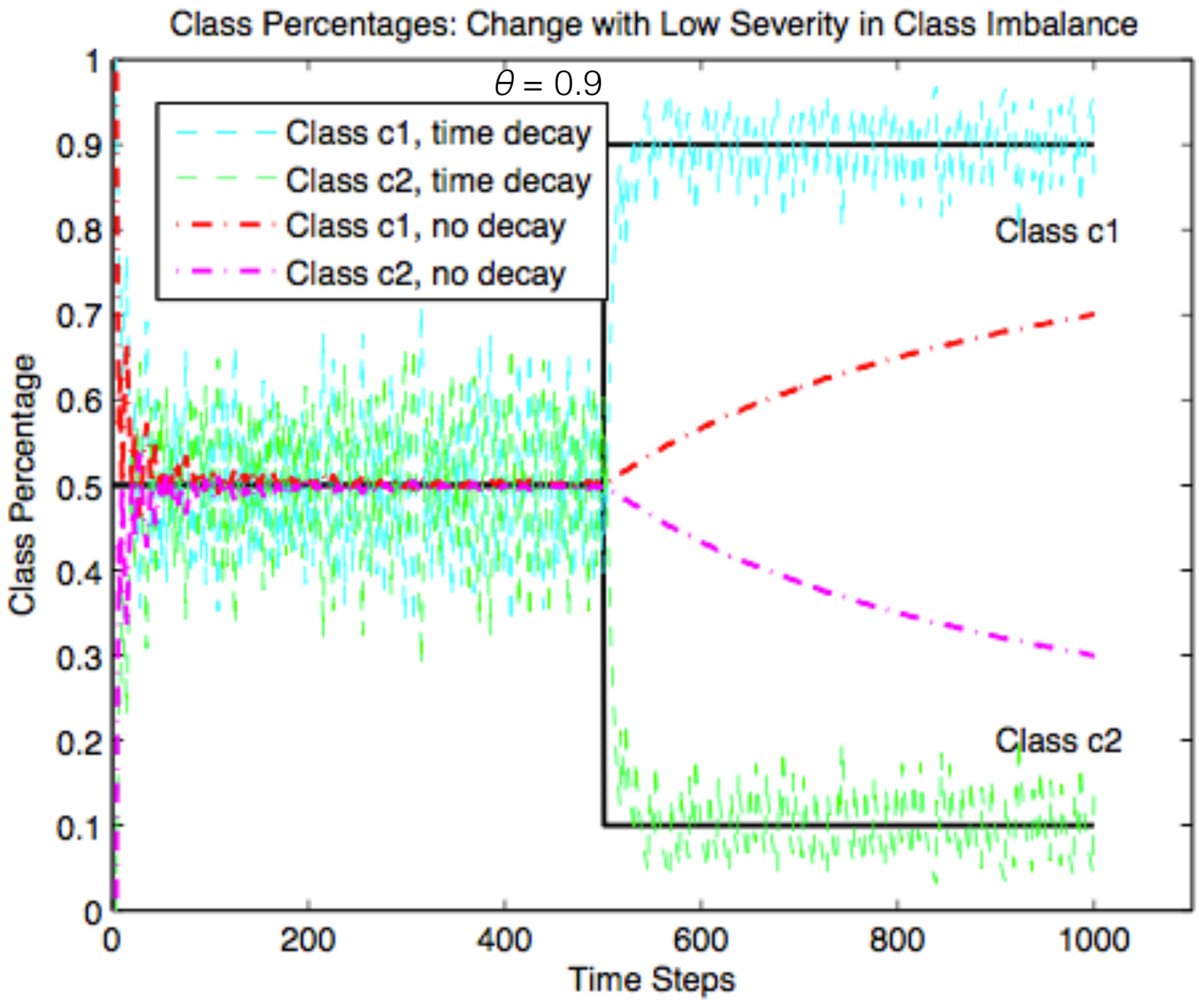
where $[(x_t, c_k)] = 1$ if true class label of x_t is c_k, and 0 otherwise; and θ is a decay factor.

S. Wang, L. Minku and X. Yao. A Learning Framework for Online Class Imbalance Learning, IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), SSCI, 2013.

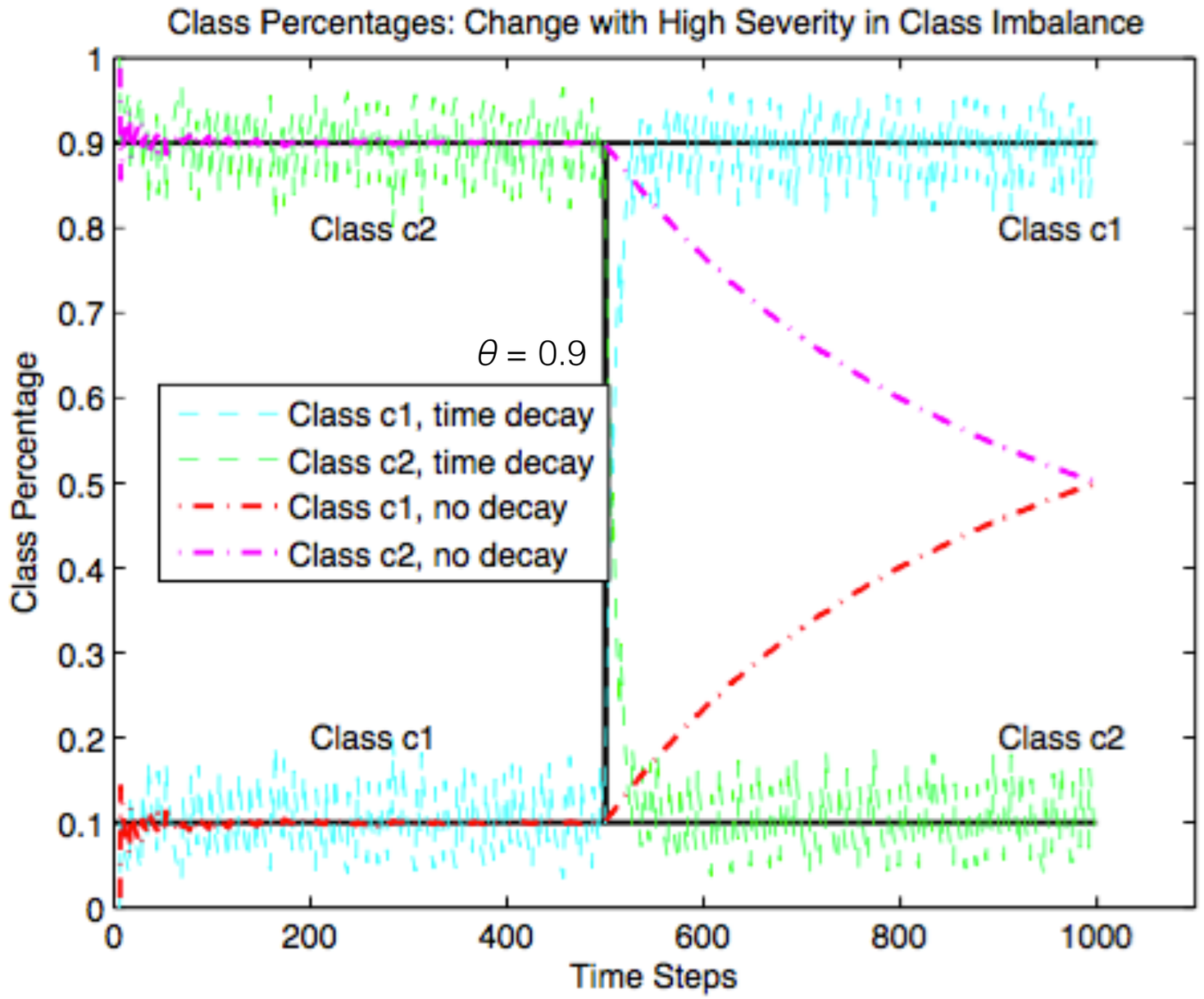
Monitoring Class Size -- No Change



Monitoring Class Size -- Change With Low Severity



Monitoring Class Size -- Change With High Severity



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 $R_i - R_j > \delta_2$

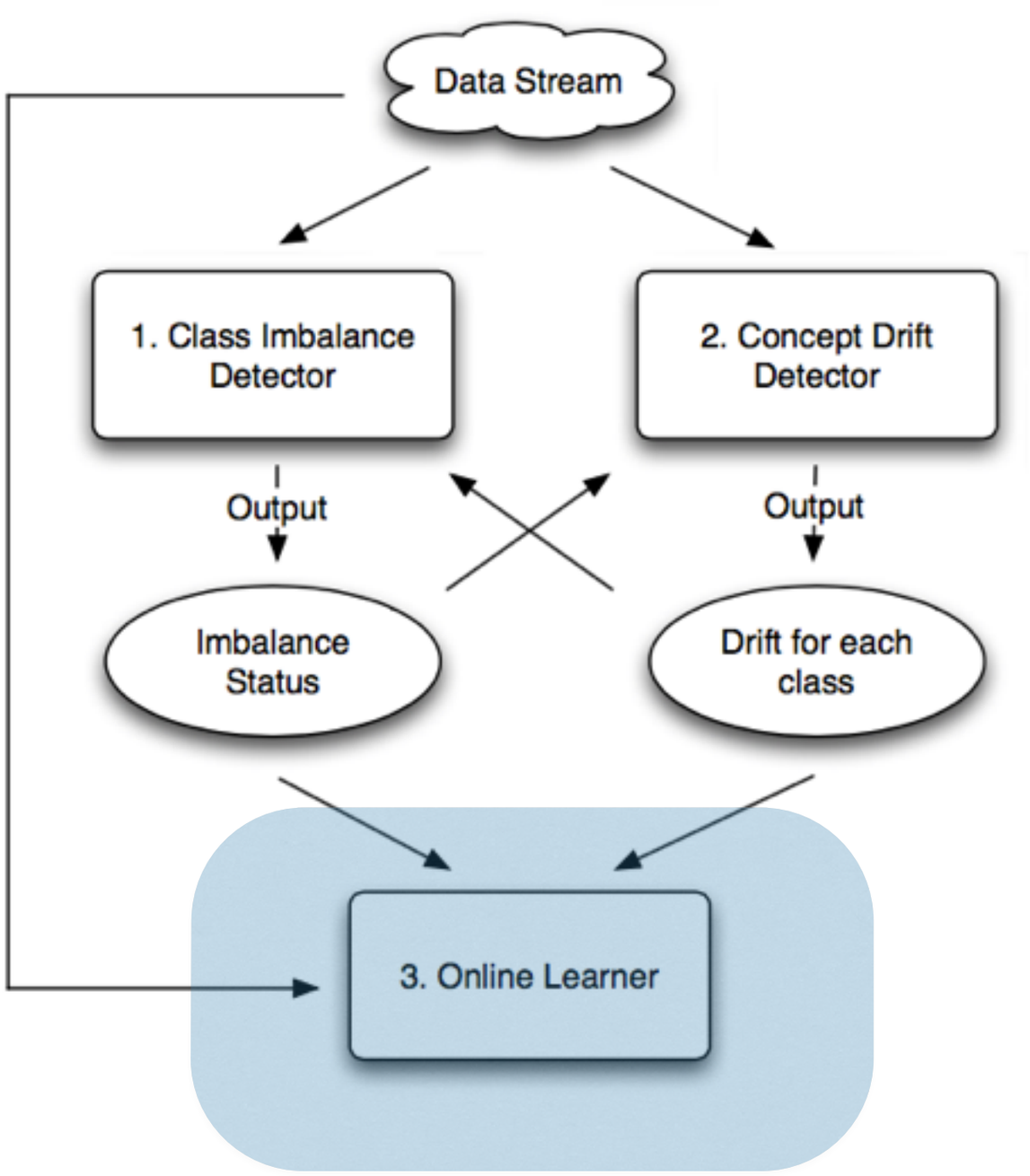
Then, class c_i is the majority; class c_j is the minority.

Time-decayed recall (TP / P):

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

where [x <-- 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 η' is a decay factor.

Online Class Imbalance Learning Framework



Resampling-Based Ensemble Approaches

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

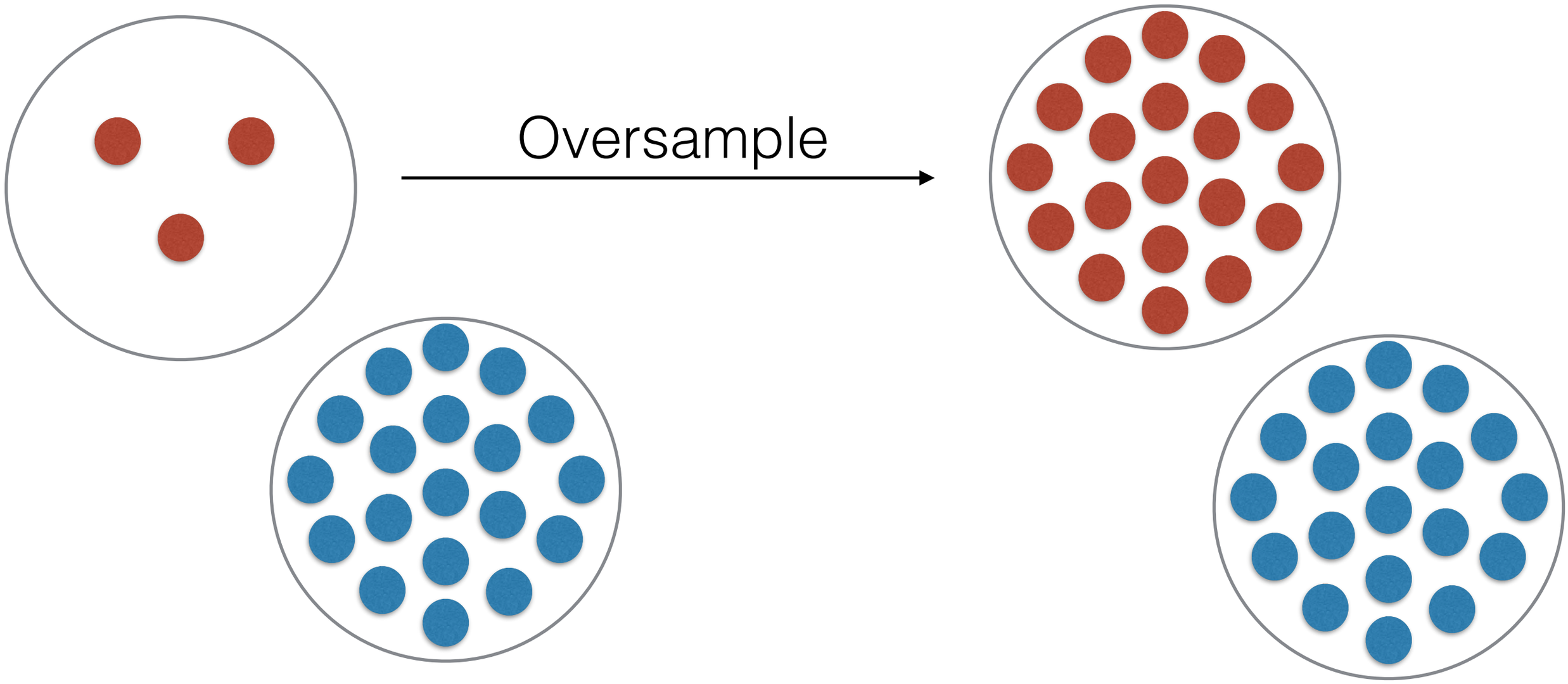
Online Bagging:

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

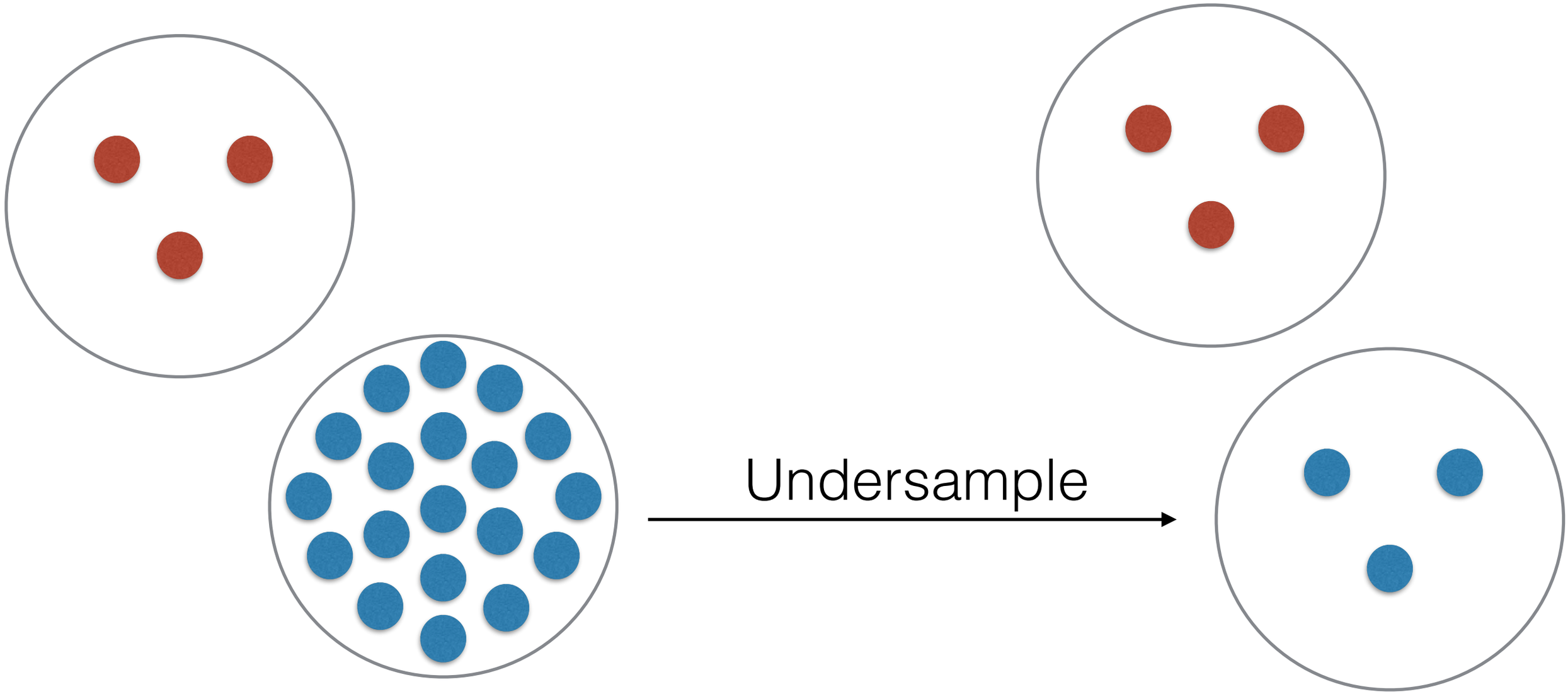
for each base learner f_m $(m = 1, 2, \ldots, M)$ **do**
  set $K \sim Poisson(\lambda)$
  update f_m K times
end for

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

Oversampling the Minority Class



Undersampling the Majority Class



Resampling-Based Ensemble Approaches

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

for each base learner f_m $(m = 1, 2, \ldots, M)$ **do**

if $y_t = +1$ and $\begin{cases} w_+^{(t)} < w^{(t)} & \text{for OOB} \\ w_+^{(t)} > w_-^{(t)} & \text{for UOB} \end{cases}$

set $K \sim Poisson\left(w_-^{(t)}/w_+^{(t)}\right)$

else if $y_t = -1$ and $\begin{cases} w_-^{(t)} < w_+^{(t)} & \text{for OOB} \\ w_-^{(t)} > w_+^{(t)} & \text{for UOB} \end{cases}$

set $K \sim Poisson\left(w_+^{(t)}/w_-^{(t)}\right)$

else

set $K \sim Poisson\,(1)$

end if

update f_m K times

end for

+1 is a "minority"

+1 is a "majority"

undersample (λ < 1)

-1 is a "minority"

-1 is a "majority"

undersample (λ < 1)

no resampling as yt 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

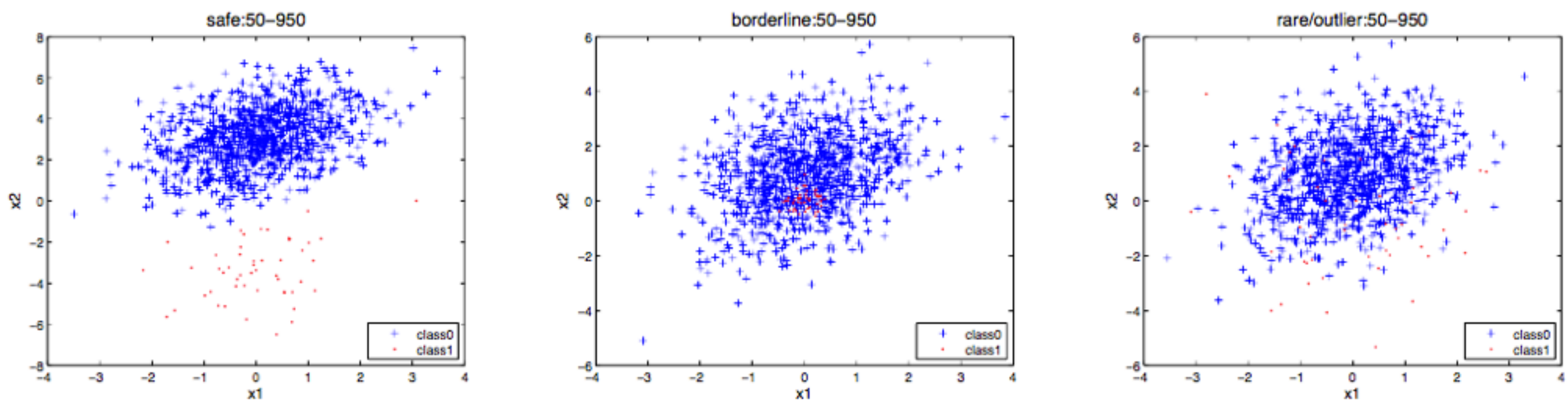
Research questions:

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.



(a) Safe (b) Borderline (c) Rare/outlier

K. Napierala and J. Stefanowski, "Identification of different types of minority class examples in imbalanced data," Hybrid Artificial Intelligent Systems, vol. 7209, pp. 139–150, 2012.

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:

- 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: 94 / 94 = 100%
minority: 0 / 6 = 0%

Evaluating Performance in Class Imbalance Learning

G-mean: $\sqrt{TP/P \cdot TN/N}$

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% * 0% = 0

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

\sqrt{100% * 50%} = ~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.

TABLE 3: The *final-step minority-class recall* and statistical test results from tree-based OOB, UOB and OB.

Data		OOB	UOB	OB
Distribute	IR			
safe	30%	0.970±0.002 ***(0.00000)***	0.973±0.001	0.969±0.001 ***(0.00000)***
	20%	0.981±0.003 ***(0.00000)***	0.964±0.002	0.964±0.004 ***(0.00510)***
	10%	0.912±0.007 ***(0.00000)***	0.936±0.005	0.905±0.007 ***(0.00000)***
	5%	0.840±0.000 ***(0.00000)***	0.917±0.013	0.876±0.019 ***(0.00000)***
borderline	30%	0.636±0.013 ***(0.00000)***	0.774±0.007	0.462±0.015 ***(0.00000)***
	20%	0.577±0 ***(0.0000***		
	10%	0.424±0 ***(0.0000***		
	5%	0.225±0 ***(0.0000***		
rare/ outlier	30%	0.197±0.016 ***(0.00000)***	0.662±0.033	0.033±0.004 ***(0.00000)***
	20%	0.395±0.015 ***(0.00000)***	0.755±0.014	0.142±0.015 ***(0.00000)***
	10%	0.195±0.024 ***(0.00000)***	0.699±0.014	0.195±0.024 ***(0.00000)***
	5%	0.310±0.010 ***(0.00000)***	0.519±0.021	0.008±0.009 ***(0.00000)***
Gearbox		0.045±0.009 ***(0.00000)***	0.446±0.041	0.000±0.000 ***(0.00000)***
Smart Building		0.430±0.004 ***(0.00000)***	0.764±0.011	0.234±0.103 ***(0.00000)***

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

Both OOB and UOB, and particularly UOB, greatly improve minority class recall over OB.

mance of all proaches ates from safe to rare, but OB suffers more.

Gmean is also better.

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.

  A. Ghazikhani, R. Monsefi, and H. S. Yazdi, “Recursive least square perceptron model for non-stationary and imbalanced data stream classification,” Evolving Systems, vol. 4, no. 2, pp. 119–131, 2013.
- WOS-ELM = Weighted Online Sequential Extreme Learning Machine.

  B. Mirza, Z. Lin, and K.-A. Toh, “Weighted online sequential extreme learning machine for class imbalance learning,” Neural Processing Letters, vol. 38, no. 3, pp. 465–486, 2013.

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

Minority class recall at the last step of the training

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

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

Data		OOB	UOB	RLSACP	WOS-ELM
Distribute	IR				
safe	30%	0.972±0.001	0.926±0.007 ***(0.00000)***	0.493±0.345 ***(0.00000)***	0.065±0.066 ***(0.00000)***
	20%	0.970±0.001	0.907±0.010 ***(0.00000)***	0.548±0.365 ***(0.00000)***	0.036±0.077 ***(0.00000)***
	10%	0.957±0.002	0.842±0.025 ***(0.00000)***	0.593±0.304 ***(0.00000)***	0.146±0.135 ***(0.00000)***
	5%	0.933±0.000	0.776±0.040 ***(0.00000)***	0.123±0.105 ***(0.00000)***	0.160±0.181 ***(0.00000)***
borderline	30%	0.586±0.007	0.515±0.022 ***(0.00000)***	0.515±0.079 ***(0.00000)***	0.231±0.137 ***(0.00000)***
	20%	0.537±0.010	0.426±0.050 ***(0.00000)***	0.475±0.086 ***(0.00008)***	0.348±0.138 ***(0.00000)***
	10%	0.500±0.009	0.374±0.096 ***(0.00000)***	0.467±0.123 (0.02940)	0.189±0.113 ***(0.00000)***
	5%	0.104±0.061	0.447±0.060 ***(0.00000)***	0.183±0.055 ***(0.00000)***	0.306±0.172 ***(0.00000)***
rare/ outlier	30%	0.399±0.016	0.463±0.026 ***(0.00000)***	0.513±0.021 ***(0.00000)***	0.476±0.056 ***(0.00000)***
	20%	0.561±0.007	0.425±0.060 ***(0.00000)***	0.482±0.093 ***(0.00000)***	0.254±0.228 ***(0.00000)***
	10%	0.598±0.011	0.447±0.094 ***(0.00000)***	0.516±0.153 (0.09300)	0.163±0.170 ***(0.00000)***
	5%	0.416±0.004	0.450±0.081 ***(0.00000)***	0.316±0.047 ***(0.00000)***	0.162±0.208 ***(0.00000)***
Gearbox		0.077±0.047	0.459±0.055 ***(0.00000)***	0.189±0.063 ***(0.00000)***	0.289±0.022 ***(0.00000)***
Smart Building		0.243±0.027	0.485±0.020 ***(0.00000)***	0.220±0.081 ***(0.00033)***	0.527±0.004 ***(0.00000)***

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.

OC

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?

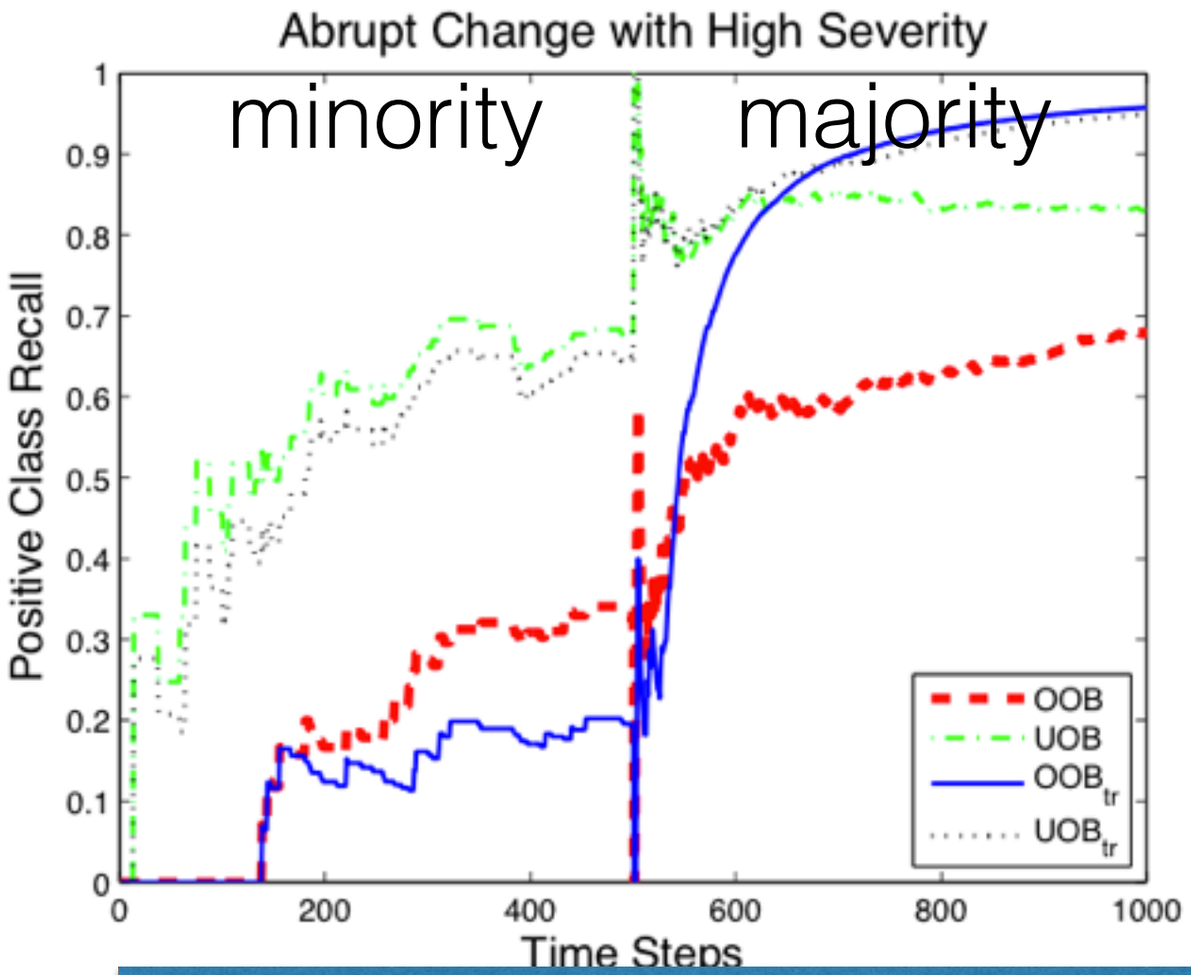
Comparisons:

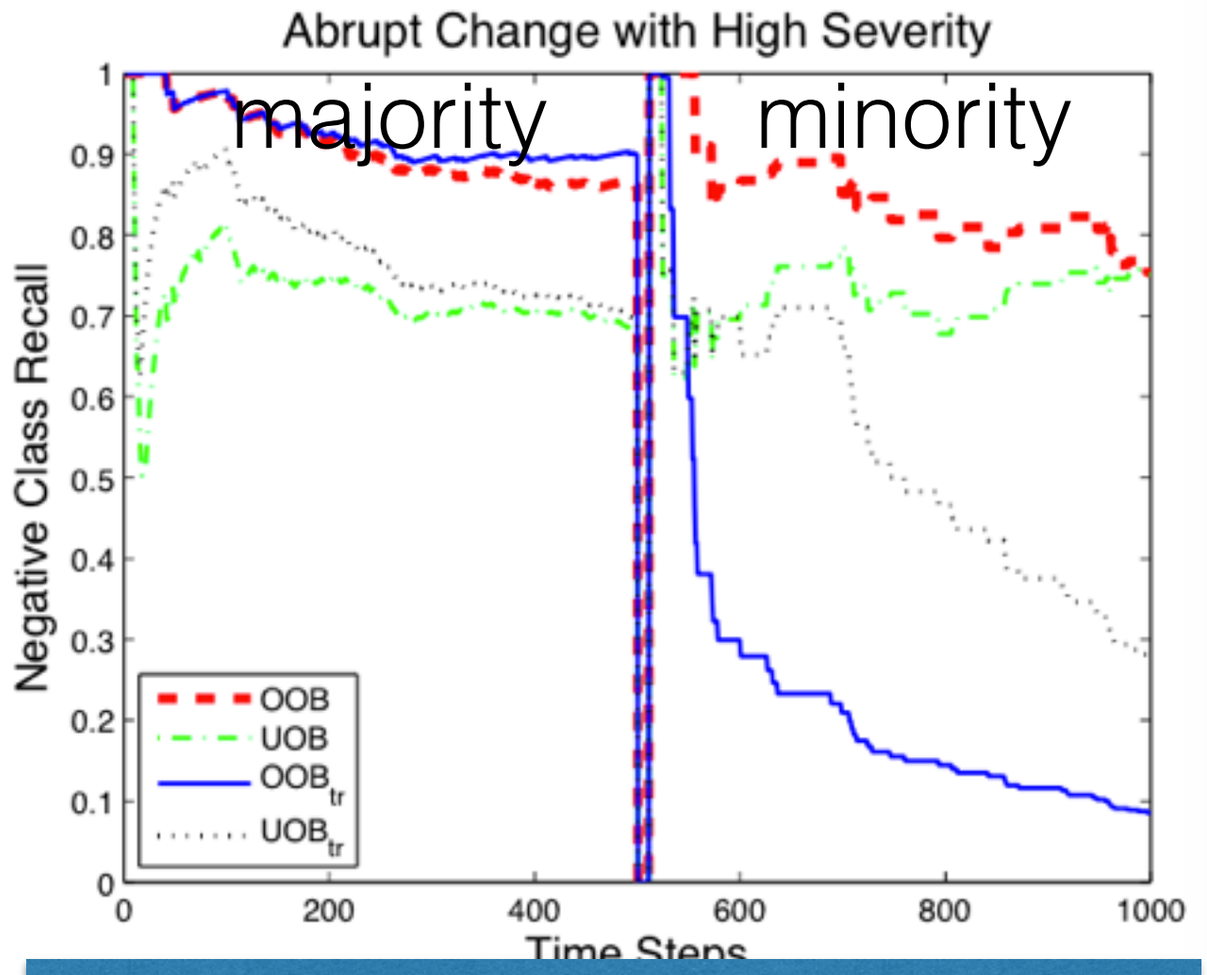
- OOB and UOB are compared with traditional methods of updating the class size (OOBtr and UOBtr).
- OOBtr and UOBtr consider all examples so far equally when determining the class size, i.e., they use no decay function.

Base learners:

- All ensembles use 50 Hoeffding trees.

RQ1: How does the time-decayed metric used in OOB and UOB help handling imbalance change?





Time-decayed metric is important.
OOB is more robust to changes than UOB.

the change.

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

Average G-mean after change

Data	Change		OOB	UOB	RLSACP	WOS-ELM
Gaussian	H	A	0.532±0.034	0.019±0.013 ***(0.00000)***	0.366±0.096 ***(0.00000)***	0.090±0.125 ***(0.00000)***
	H	G	0.360±0.002	0.359±0.011 ***(0.00440)***	0.388±0.089 (0.04490)	0.037±0.095 ***(0.00000)***
	L	A	0.795±0.003	0.521±0.025 ***(0.00000)***	0.439±0.210 ***(0.00000)***	0.022±0.074 ***(0.00000)***
	L	G	0.811±0.001	0.758±0.015 ***(0.00000)***	0.613±0.118 ***(0.00000)***	0.015±0.056 ***(0.00000)***
Gearbox	H	A	0.293±0.009	0.201±0.032 ***(0.00000)***	0.438±0.012 ***(0.00000)***	0.000±0.000 ***(0.00000)***
	H	G	0.000±0.000	0.364±0.119 ***(0.00000)***	0.140±0.126 ***(0.00000)***	0.000±0.000 (NaN)
	L	A	0.452±0.014	0.479±0.012 ***(0.00000)***	0.378±0.031 ***(0.00000)***	0.000±0.000 ***(0.00000)***
	L	G	0.408±0.032	0.432±0.023 ***(0.00000)***	0.055±0.095 ***(0.00000)***	0.000±0.000 ***(0.00000)***
Smart Building	H	A	0.276±0.028	0.261±0.052 (0.04740)	0.135±0.156 ***(0.00000)***	0.123±0.008 ***(0.00000)***
	H	G	0.000±0.000	0.140±0.091 (0.12090)	0.157±0.188 ***(0.00000)***	0.000±0.000 (NaN)
	L	A	0.451±0.019	0.435±0.016 ***(0.00000)***	0.228±0.144 ***(0.00000)***	0.000±0.000 ***(0.00000)***
	L	G	0.443±0.023	0.485±0.018 ***(0.00000)***	0.080±0.078 ***(0.00000)***	0.000±0.000 ***(0.00000)***

Either OOB [illegible] most cases.

Similar res[illegible]UOB use a single MLP, suggesting that resampling plays a key role.

The proposed approaches are competitive in comparison to previous approaches.

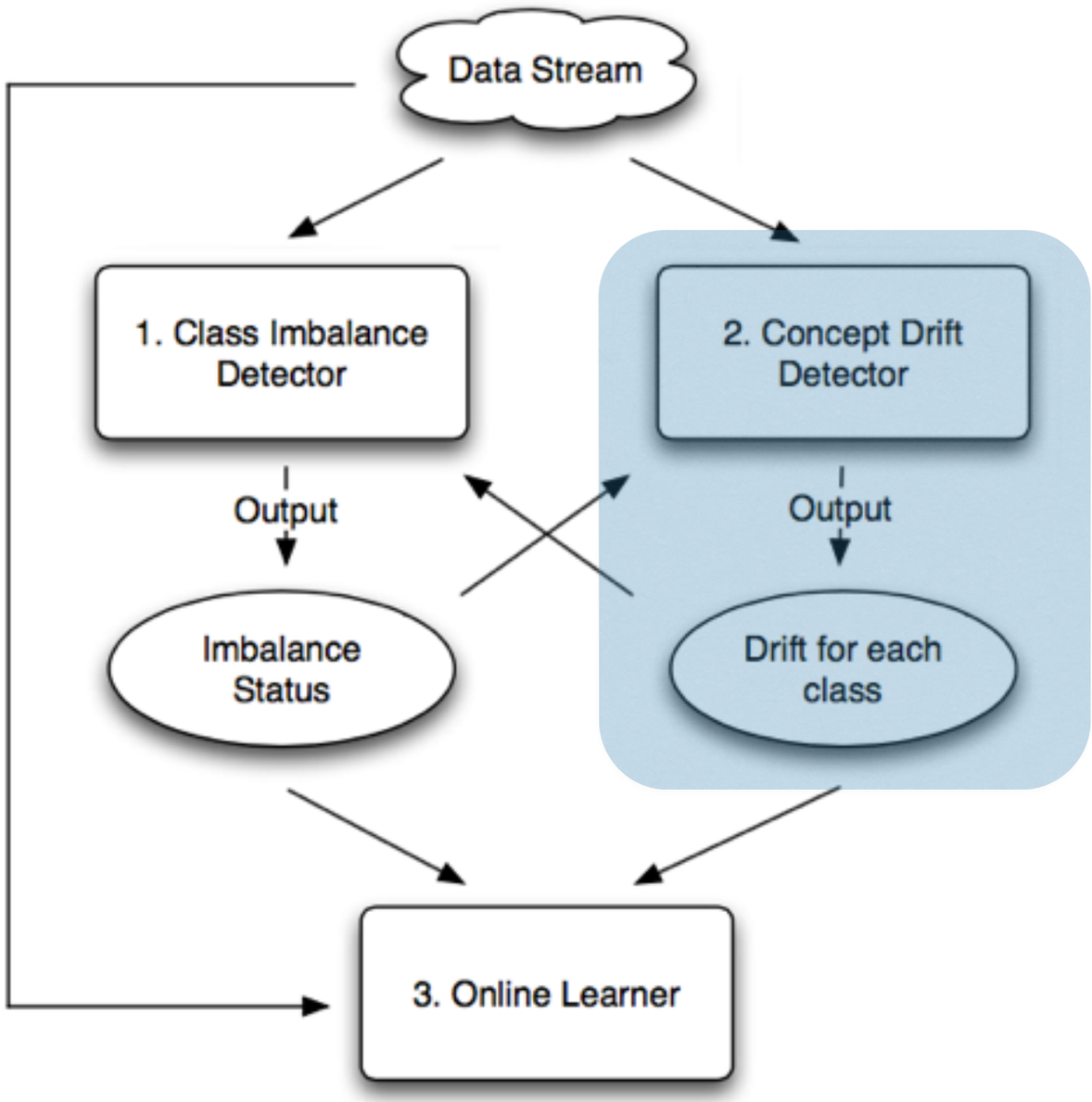
Ensemble of OOB and UOB

- 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



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

$a_1 + a_2 <= \theta$ --> *class 0*

$a_1 + a_2 > \theta$ --> *class 1 (minority)*

STAGGER

Boolean equations involving size (S, M, L), shape (circle, triangle, rectangle) and colour (R, G, B).

False --> class 0

True --> class 1 (minority)

iNemo

Multi-sensing platform (accelerometers, gyroscopes and magnetometers with pressure and temperature). An offset θ is added to the gyroscope signal to simulate faults.

Non-faulty --> class 0

Faulty --> class 1 (minority)

Data Sets

Concept Drift

Concept	SEA	STAGGER	iNemo
Old	$\theta = 13$	size=small $\cap$ color=red	$\theta = 500$
New Low Severity	$\theta = 10$	(size=small $\cap$ color=red) $\|\|$ (color=green $\cup$ shape=square)	$\theta = 500 \,\|\|\, \theta = -500$
New High Severity	$\theta = 7$	color=green $\cup$ shape=square	$\theta = -500$

Class Imbalance Status

	p_{c0}	p_{c1}
Old	0.9	0.1
New Low Severity	0.5	0.5
New High Severity	0.1	0.9

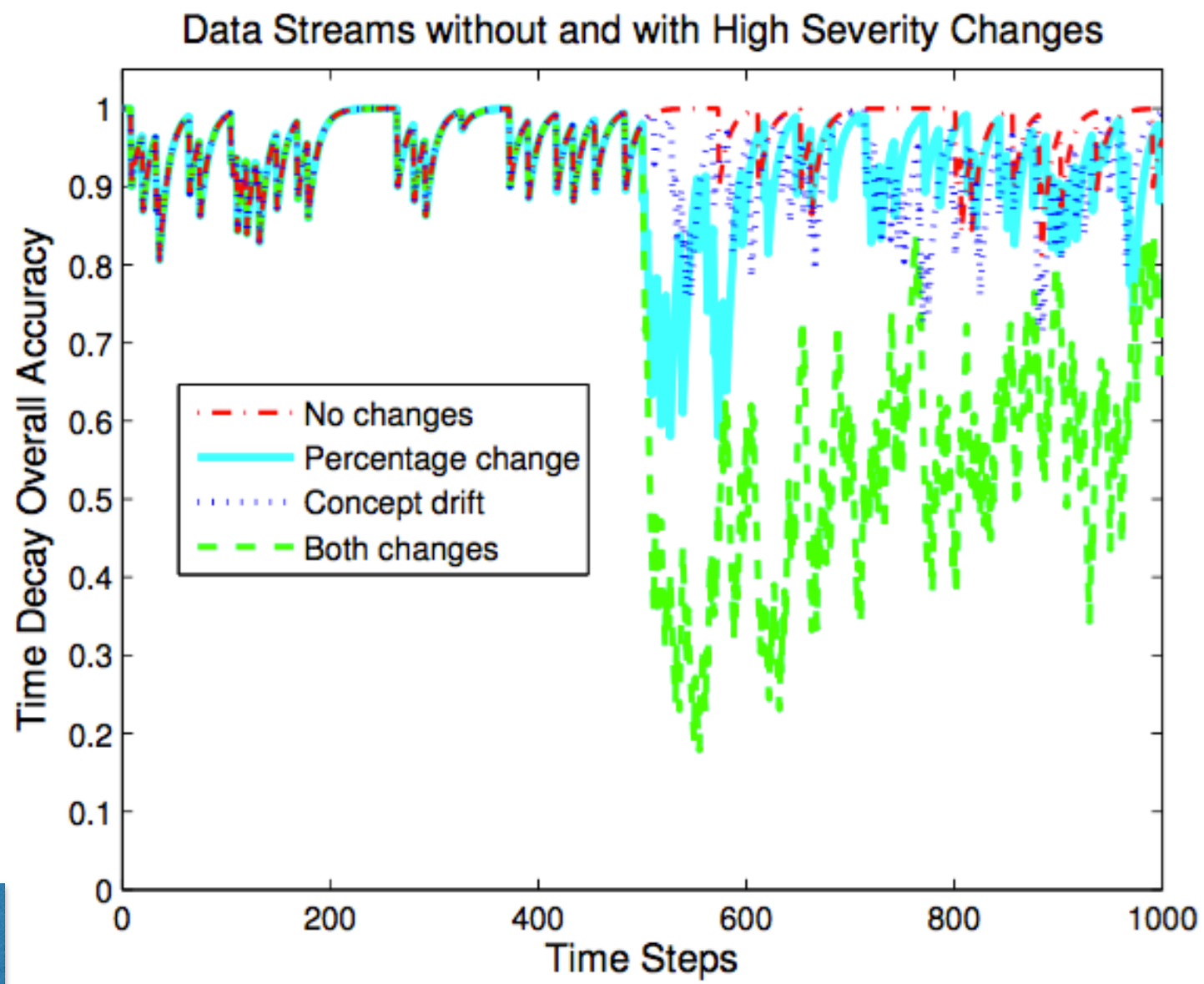
Data stream size: 1000 examples
Abrupt change at 501

Case	Class imbalance	Concept drift
Case 1	–	–
Case 2	High	–
Case 3	Low	–
Case 4	–	High
Case 5	–	Low
Case 6	High	High
Case 7	Low	Low

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.

Accuracy is not adequate for detecting concept drift in class-imbalance environments!

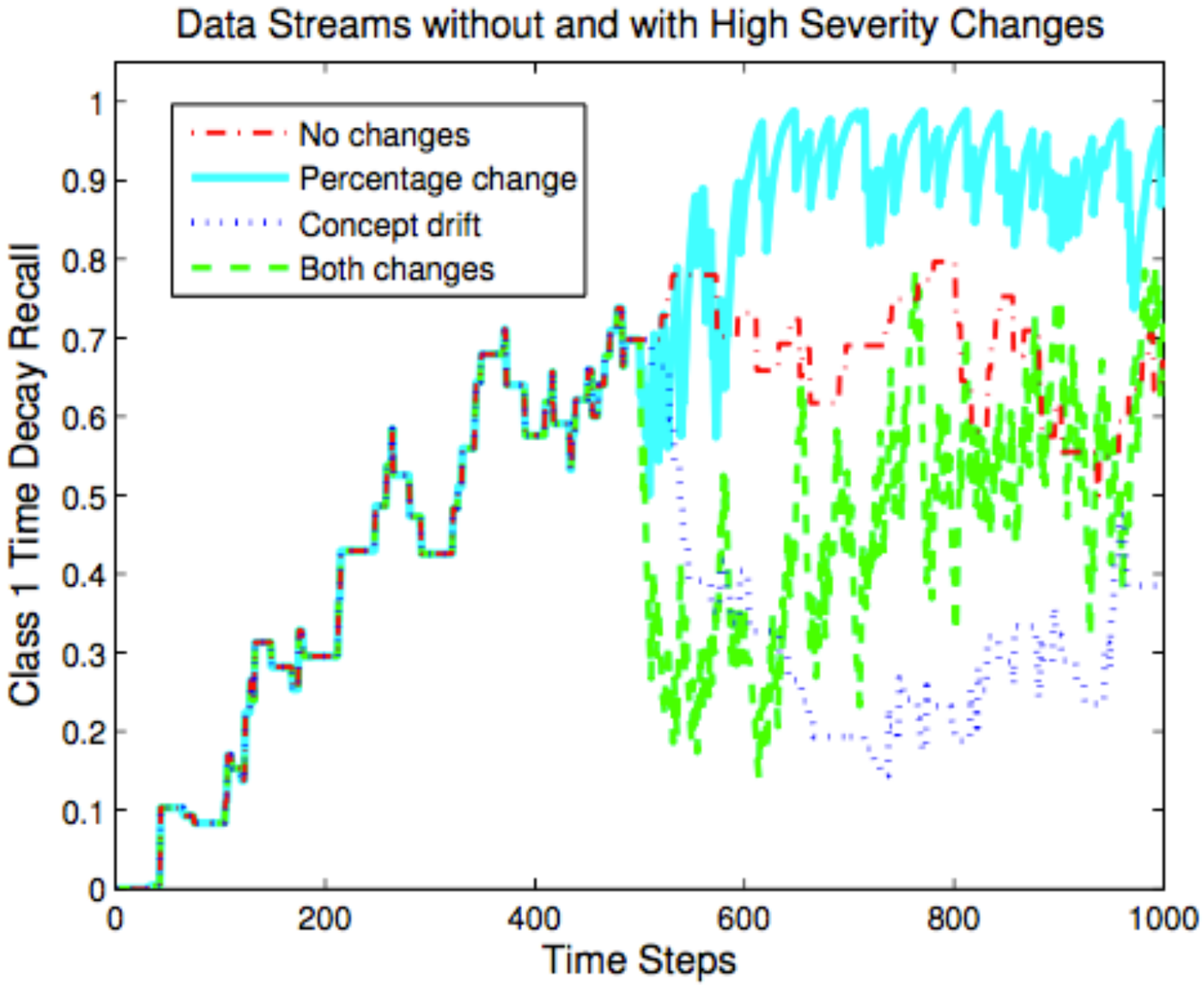


SEA: time-decayed overall accuracy under the scenarios of high severity changes produced by DT-based OB.

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_{max} and s_{max} to remember when $p_i + s_i$ reaches its maximum, and check the following conditions:

- Warning level $p_i - s_i \leq p_{max} - 2s_{max}$ 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 $p_i - s_i \leq p_{max} - 3s_{max}$ 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.

WANG, S.; MINKU, L.; GHEZZI, D.; CALTABIANO, D.; TINO, P.; YAO, X. . "Concept Drift Detection for Online Class Imbalance Learning", Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), 10 pages, August 2013, doi: 10.1109/IJCNN.2013.6706768

Experimental Study

Objectives:

- Check the suitability of DDM-OCI.

Comparisons:

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

  J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” Advances in Artificial Intelligence, vol. 3171, pp. 286–295, 2004.

Base learners:

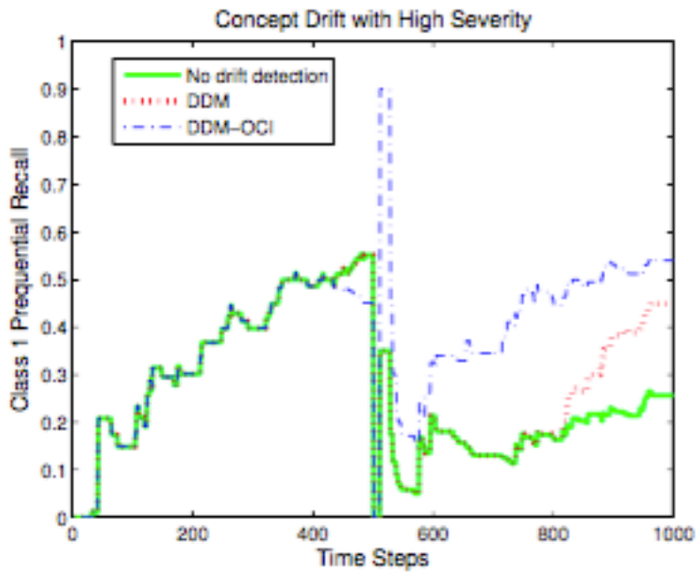
- Online bagging with 50 DTs
- Online bagging with 50 NB

Results

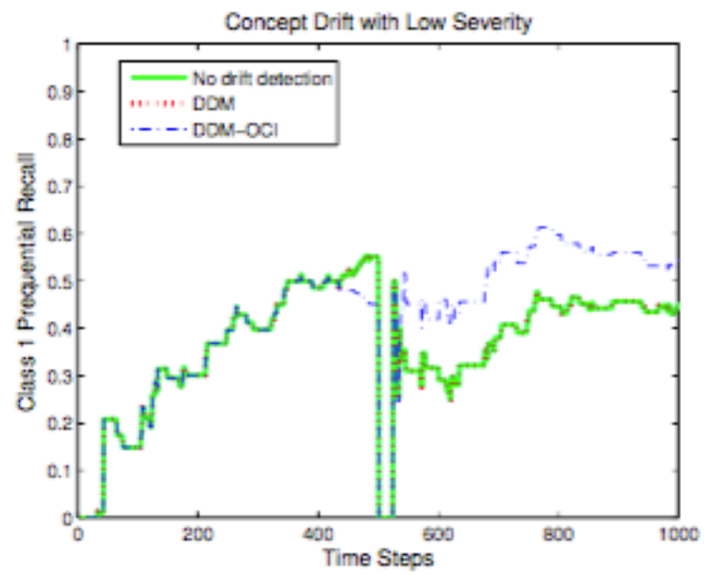
No drift detection performs the worse

DDM-OCI reacts faster.

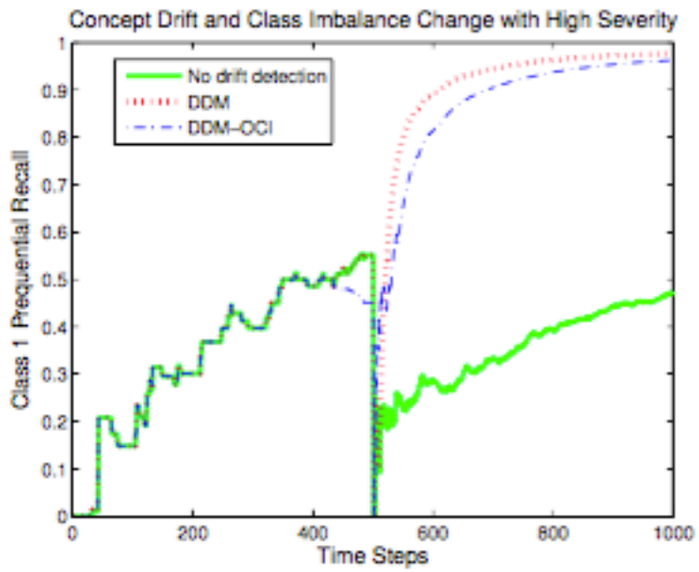
DDM-OCI suffers from false alarms.



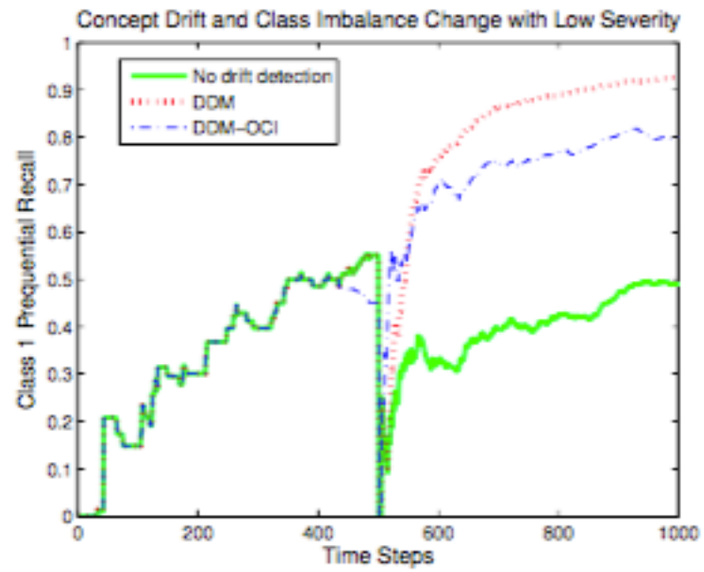
(a) Case 4: concept drift with high severity



(b) Case 5: concept drift with low severity



(c) Case 6: concept drift and class imbalance change with high severity



(d) Case 7: concept drift and class imbalance change with low severity

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.

	SEA				STAGGER				iNemo			
Case	4	5	6	7	4	5	6	7	4	5	6	7
DDM	1	0	2	1	1	2	1	1	1	0	1	1
DDM-OCI	2	2	2	6	2	2	1	4	1	2	1	2

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

  DYER K., CAPO R., POLIKAR, R., “COMPOSE: A Semi-Supervised Learning Framework for Initially Labeled Non-Stationary Streaming Data”, IEEE Transactions on Neural Networks and Learning Systems, 25(1):12-26, 2014.

- Scalable memories

  ZLIOBAITE, I.; BIFET, A.; GABER; M.; GABRYS, B.; GAMA, J.; MINKU, L.; MUSIAL, K. . "Next Challenges for Adaptive Learning Systems.", ACM SIGKDD Explorations Newsletter, 14(1): 48-55, 2012.

References

OZA, N. and RUSSEL, S., “Experimental comparisons of online and batch versions of bagging and boosting,” in Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, p. 359–364, 2001.

MINKU, L. L.; White, A.; YAO, X. . "The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift", IEEE Transactions on Knowledge and Data Engineering 22(5), 2010.

MINKU, L. L.; YAO, X. . "DDD: A New Ensemble Approach for Dealing with Concept Drift", IEEE Transactions on Knowledge and Data Engineering 24(4): 2012.

BAENA-GARCIA, M., DEL CAMPO-AVILA, J., FIDALGO, R. and BIFET, A. “Early drift detection method”, Proc. of the 4th ECML PKDD International Workshop on Knowledge Discovery From Data Streams, Berlin, 2006, pp. 77–86.

KOLTER, J. and MALOOF, M., “Dynamic weighted majority: An ensemble method for drifting concepts,” Journal of Machine Learning Research, 8:2755–2790, 2007.

WANG, S.; MINKU, L. L.; YAO, X. . "Resampling-Based Ensemble Methods for Online Class Imbalance Learning", IEEE Transactions on Knowledge and Data Engineering, 14p., 2014, doi: 10.1109/TKDE.2014.2345380 (in press)

GHAZIKHANI, A., MONSEFI, R., and YAZDI, H. S., “Recursive least square perceptron model for non-stationary and imbalanced data stream classification,” Evolving Systems, 4(2):119–131, 2013.

MIRZA, B., LIN, Z., and TOH, K.-A., “Weighted online sequential extreme learning machine for class imbalance learning,” Neural Processing Letters, 38(3): 465–486, 2013.

WANG, S.; MINKU, L.; YAO, X. . “A Learning Framework for Online Class Imbalance Learning”, IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), at IEEE Symposium Series on Computational Intelligence (SSCI), p. 36-45, Singapore, April 2013.

WANG, S.; MINKU, L.; GHEZZI, D.; CALTABIANO, D.; TINO, P.; YAO, X. . “Concept Drift Detection for Online Class Imbalance Learning”, Proceedings of the 2013 International Joint Conference on Neural Networks, 10 pages, August 2013

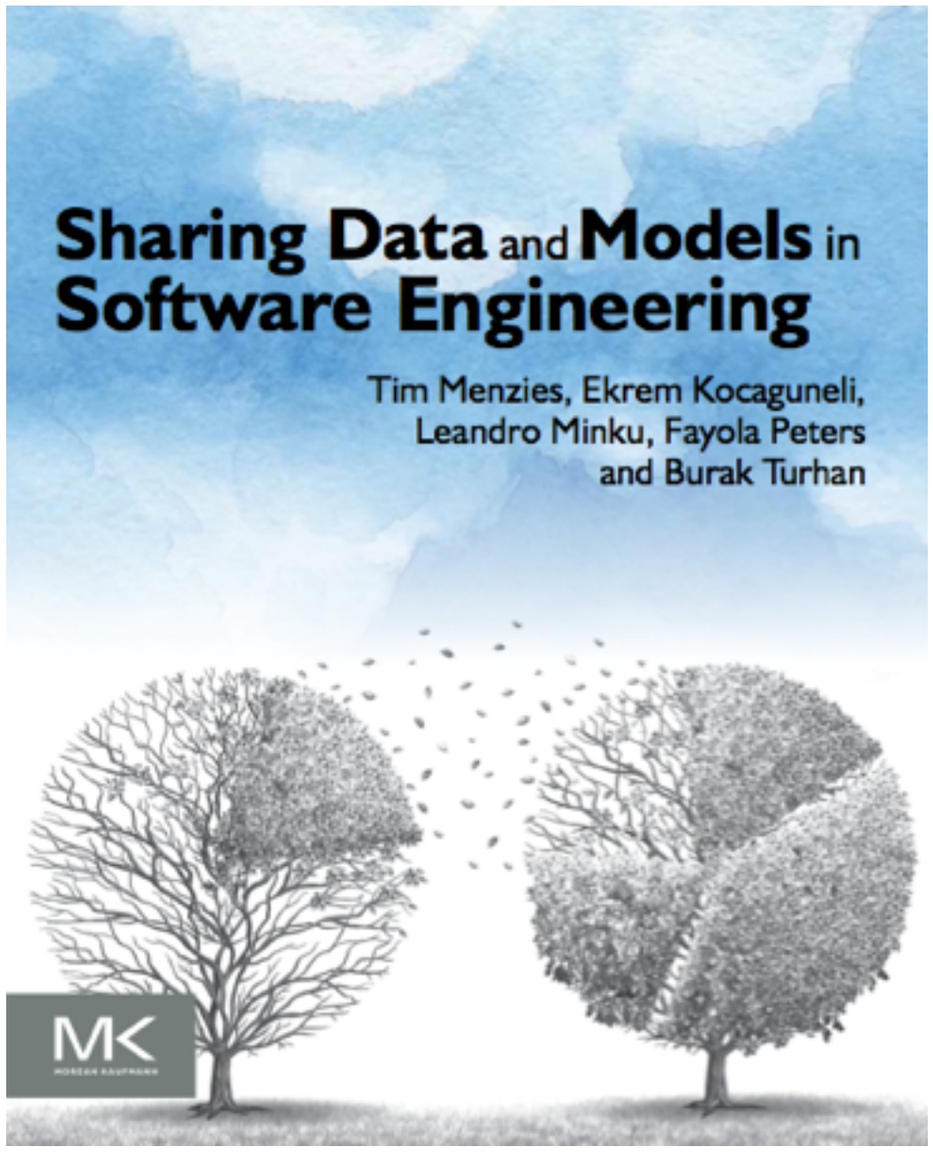
NAPIERALA, K. and STEFANOWSKI, J., “Identification of different types of minority class examples in imbalanced data,” Hybrid Artificial Intelligent Systems, 7209:139–150, 2012.

GAMA, MEDAS, CASTILLO, G., RODRIGUES,P, “Learning with drift detection,” Advances in Artificial Intelligence, 3171:286–295, 2004.

MINKU, L. L.; YAO, X. . "Can Cross-Company Data Improve Performance in Software Effort Estimation?" International Conference on Predictive Models in Software Engineering 2012.

MINKU, L. L.; YAO, X. . "How to Make Best Use of Cross-Company Data in Software Effort Estimation? International Conference on Software Engineering 2014.

Thank you!


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