Adaptive Machine Learning for Credit Card Fraud Detection

Andrea Dal Pozzolo

4/12/2015

PhD public defence

Supervisor: Prof. Gianluca Bontempi

THE *Doctiris* PROJECT

- 4 years PhD project funded by InnovIRIS (Brussels Region)
- Industrial project done in collaboration with Worldline S.A. (Brussels, Belgium)
- Requirement: 50% time at Worldline (WL) and 50% at MLG-ULB
- Goal: Improve the Data-Driven part of the Fraud Detection System (FDS) by means of Machine Learning algorithms.

innoviris.brussels
empowering research

Introduction

- This thesis is about Fraud Detection (FD) in credit card transactions.
- FD is the process of identifying fraudulent transactions **after** the payment is authorized.
- This process is typically automatized by means of a FDS.

Technical sections will be denoted by the symbol 

The Fraud Detection process

Terminal check

Correct Pin?
Sufficient Balance ?
Blocked Card?

Rejected

Predictive model

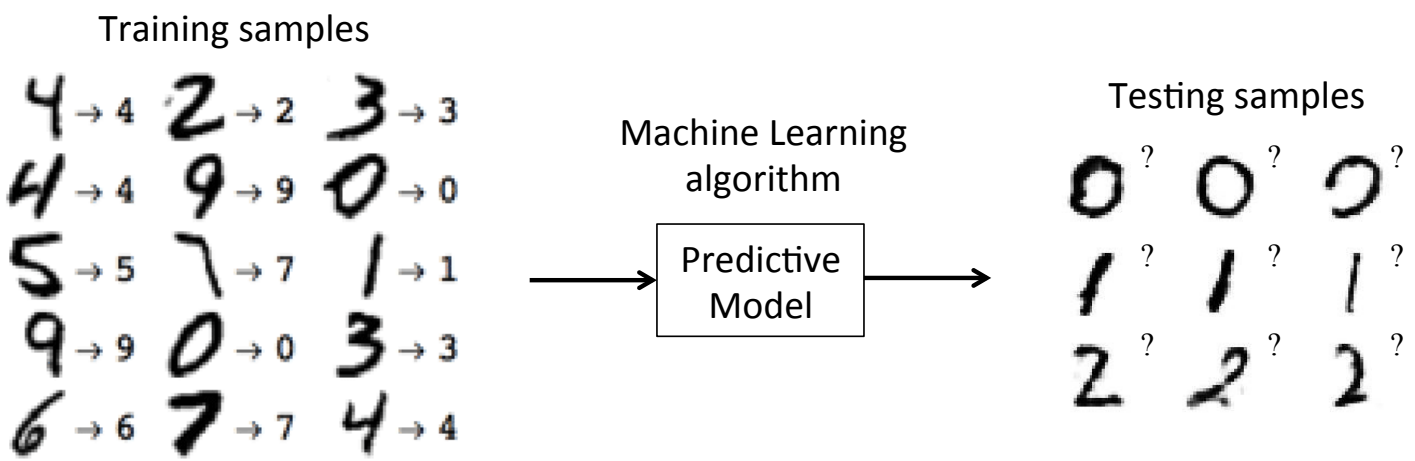
Fraud score

Feedback

Investigators

What's Machine Learning?

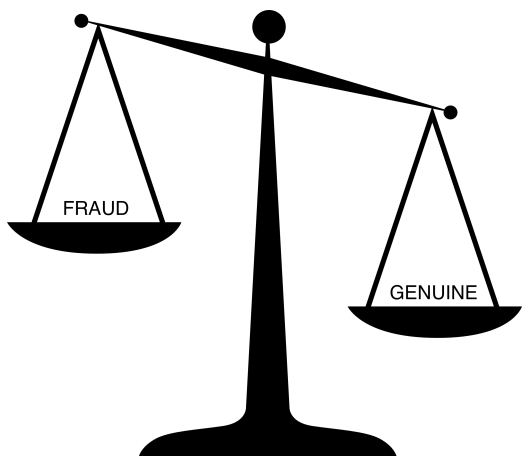
- The design of algorithms that **discover patterns** in a collection of data instances in an **automated** manner.
- The goal is to use the discovered patterns to make predictions on new data.



We let **computers learn these patterns**.

Challenges for a Data-Driven FDS

1. Unbalanced distribution (few frauds).
2. Concept Drift (CD) due to fraud evolution and change in customer behaviour.
3. Few recent supervised transactions (true class of only transactions reviewed by investigators).
4. For confidentiality reason, exchange of datasets and information is often difficult.



PCA PROJECTION

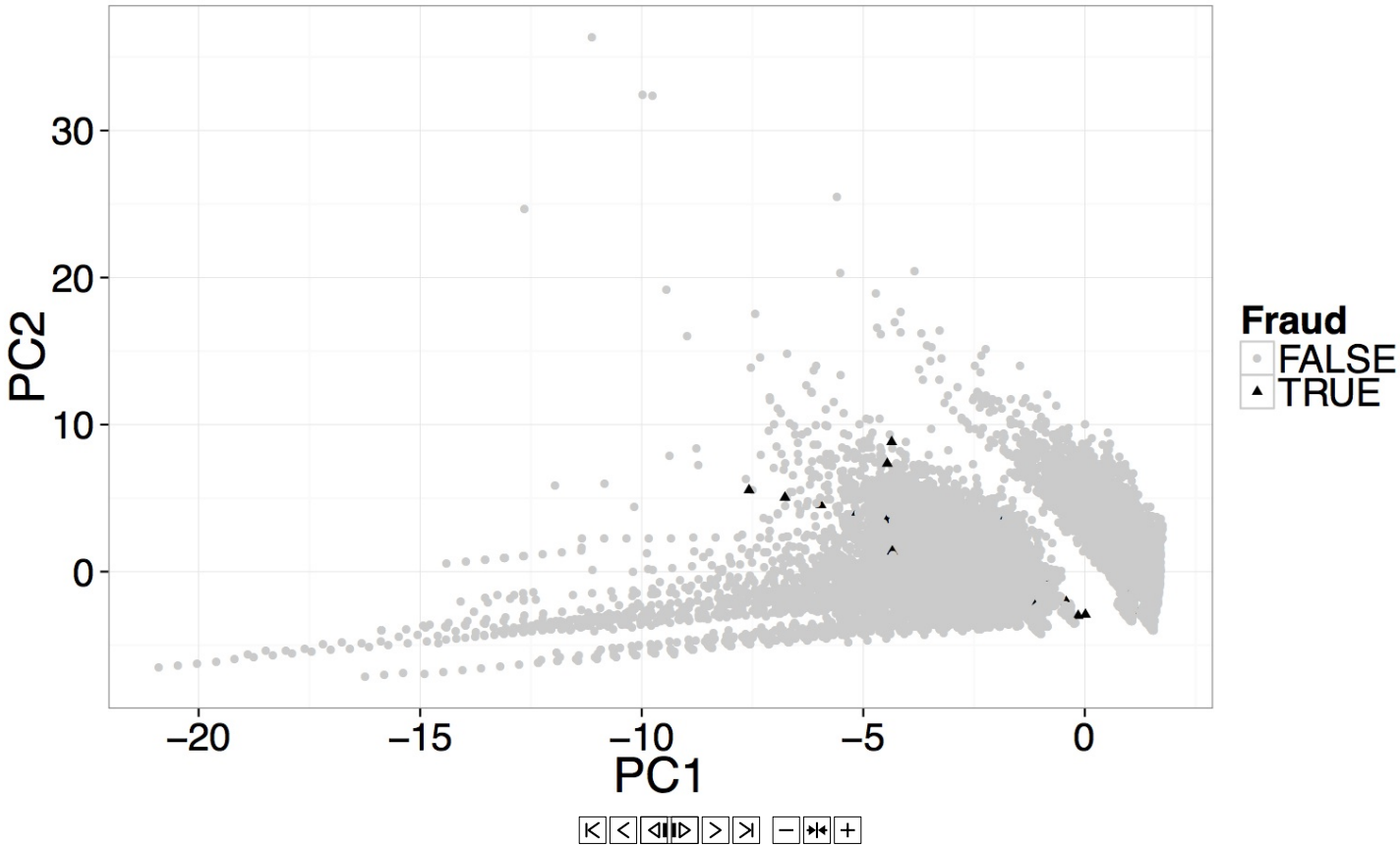


Figure : Transactions distribution over the first two Principal Components in different days.

RESAMPLING METHODS

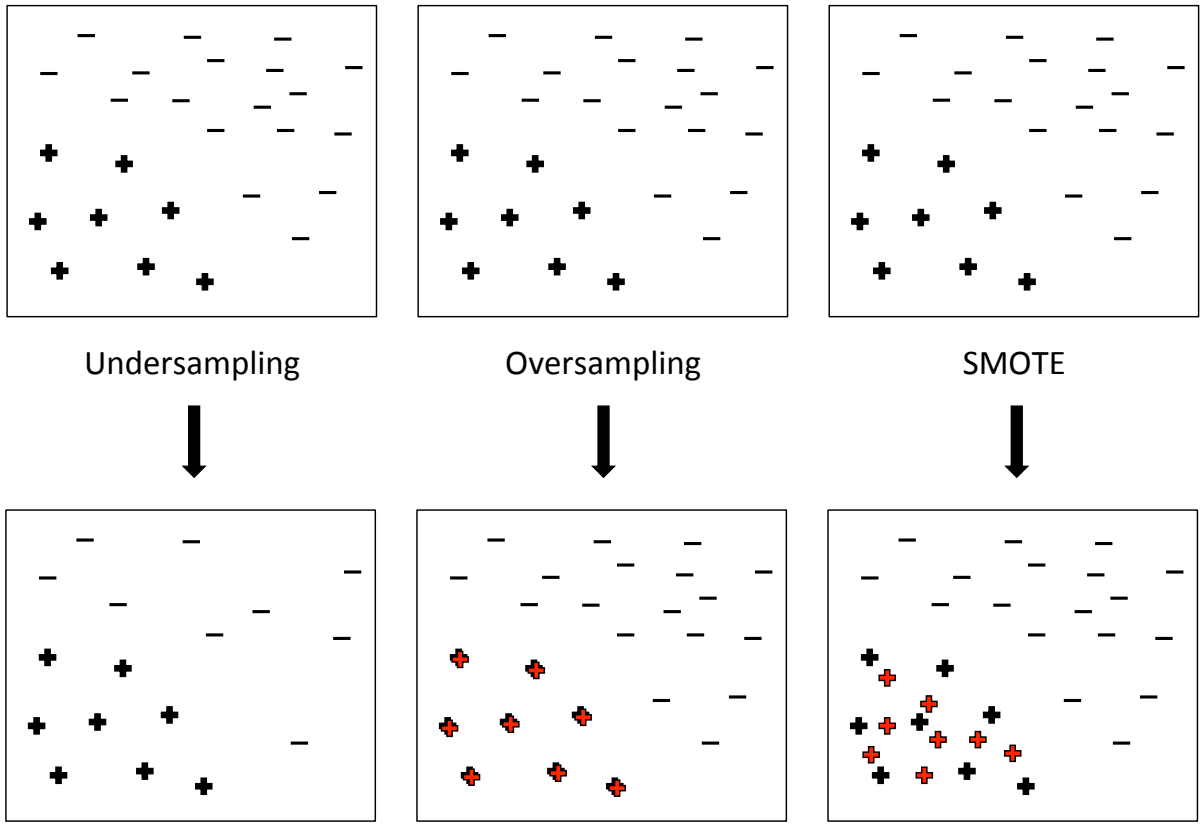


Figure : Resampling methods for unbalanced classification. The positive and negative symbols denote fraudulent and genuine transactions. In red the new observations created with oversampling methods.

Concept Drift

In data streams the concept to learn can change due to non-stationary distributions.

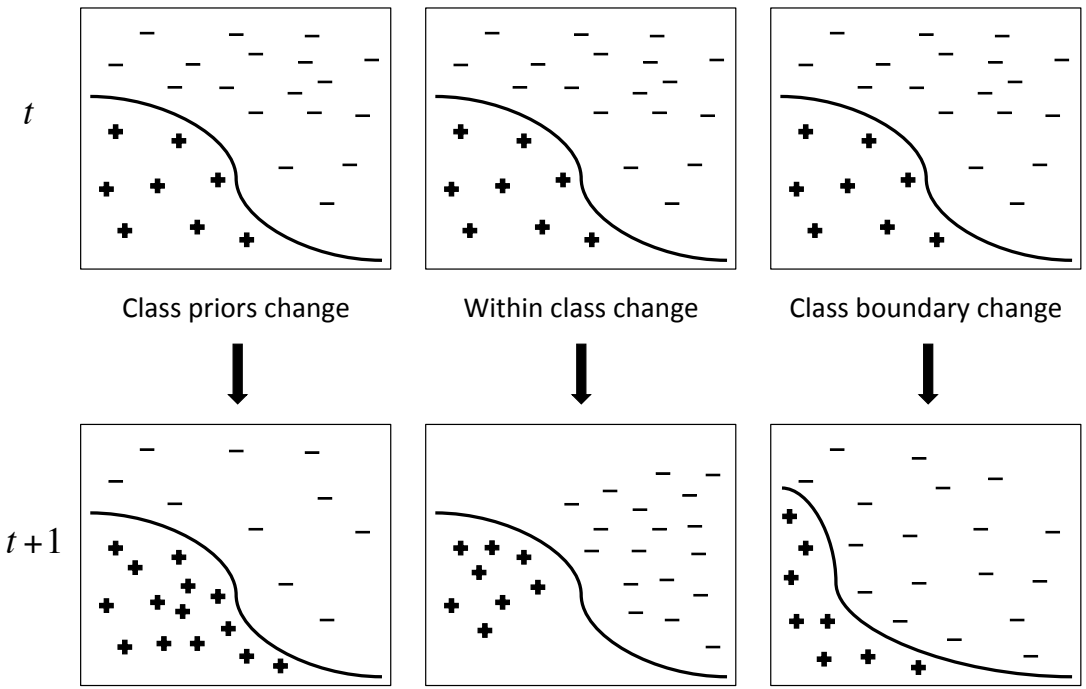


Figure : Illustrative example of different types of Concept Drifts.

Thesis Contributions

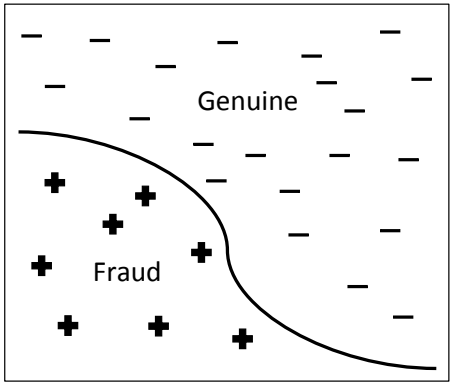
1. Learning with unbalanced distribution:
   - impact of undersampling on the accuracy of the final classifier
   - racing algorithm to adaptively select the best unbalanced strategy
2. Learning from evolving and unbalanced data streams
   - multiple strategies for unbalanced data stream.
   - effective solution to avoid propagation of old transactions.
3. Prototype of a real-world FDS
   - realistic framework that uses all supervised information and provides accurate alerts.
4. Software and Credit Card Fraud Detection Dataset
   - release of a software package and a real-world dataset.

PUBLICATIONS

- A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi.
  Credit Card Fraud Detection with Alert-Feedback Interaction.
  Submitted to IEEE TNNLS, 2015.
- A. Dal Pozzolo, O. Caelen, and G. Bontempi.
  When is undersampling effective in unbalanced classification tasks?
  In ECML-KDD, 2015.
- A. Dal Pozzolo, O. Caelen, R. Johnson and G. Bontempi.
  Calibrating Probability with Undersampling for Unbalanced Classification.
  In IEEE CIDM, 2015.
- A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi.
  Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information.
  In IEEE IJCNN, 2015.
- A. Dal Pozzolo, R. Johnson, O. Caelen, S. Waterschoot, N. Chawla, and G. Bontempi.
  Using HDDT to avoid instances propagation in unbalanced and evolving data streams.
  In IEEE IJCNN, 2014.
- A. Dal Pozzolo, O. Caelen, Y. Le Borgne, S. Waterschoot, and G. Bontempi.
  Learned lessons in credit card fraud detection from a practitioner perspective.
  Elsevier ESWA, 2014.
- A. Dal Pozzolo, O. Caelen, S. Waterschoot, G. Bontempi,
  Racing for unbalanced methods selection.
  In IEEE IDEAL, 2013

The Fraud Detection problem

- We formalize FD as a classification task $f : \mathbb{R}^n \to \{+, -\}$.
- $X \in \mathbb{R}^n$ is the input and $Y \in \{+, -\}$ the output domain.
- $+$ is the fraud (minority) and $-$ the genuine (majority) class.
- Given a classifier $\mathcal{K}$ and a training set T_N, we are interested in estimating for a new sample (x, y) the posterior probability $\mathcal{P}(y = +|x)$.



The Fraud Detection data

- WL is processing millions of transactions per day.
- When a transaction arrives it contains about 20 raw features (e.g. CARD_ID, AMOUNT, DATETIME, SHOP_ID, CURRENCY, etc.).
- Aggregate features are compute to include the cardholder historical behaviour (e.g. AVG_AMOUNT_WEEK, TIME_FROM_LAST_TRX, NB_TRX_SAME_SHOP, etc.).
- Final feature vector contains about 50 variables.

Accuracy measure of a FDS

- In a detection problem it is more important to provide accurate ranking than correct classification.
- In a realistic FDS, investigators check only a limited number of transactions.
- Hence, P_k is the most relevant measure, with k being the number of transactions checked.
- P_k is the proportion of frauds in k most risky transactions.

k	Predicted Score	True Class
1	0.99	+
2	0.98	+
3	0.98	-
4	0.96	-
5	0.95	+
6	0.92	-
7	0.91	+
8	0.84	-
9	0.82	-
10	0.79	-
11	0.76	-
12	0.75	+
13	0.7	-
14	0.67	-
15	0.64	+
16	0.61	-
17	0.58	-
18	0.55	+
19	0.52	-
20	0.49	-

Theoretical contribution

CHAPTER 4
-
UNDERSTANDING SAMPLING METHODS

Publications:

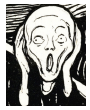
- A. Dal Pozzolo, O. Caelen, and G. Bontempi.
  When is undersampling effective in unbalanced classification tasks?
  In ECML-KDD, 2015.
- A. Dal Pozzolo, O. Caelen, R. Johnson and G. Bontempi.
  Calibrating Probability with Undersampling for Unbalanced Classification.
  In IEEE CIDM, 2015.
- A. Dal Pozzolo, O. Caelen, S. Waterschoot, G. Bontempi,
  Racing for unbalanced methods selection.
  In IEEE IDEAL, 2013

Objective of Chapter 4

In Chapter 4 we aim to

- analyse the shift in the posterior distribution due to undersampling and to study its impact on the final ranking.
- show under which condition undersampling is expected to improve classification accuracy.
- propose a method to calibrate posterior probabilities and to select the classification threshold.
- show how to efficiently select the best unbalanced strategy for a given dataset.

Effect of undersampling



- Suppose that a classifier $\mathcal{K}$ is trained on set T_N which is unbalanced.
- Let $\mathbf{s}$ be a random variable associated to each sample $(x, y) \in \mathrm{T}_N$, $\mathbf{s} = 1$ if the point is sampled and $\mathbf{s} = 0$ otherwise.
- Assume that $\mathbf{s}$ is independent of the input x given the class y (*class-dependent selection*):

$$\mathcal{P}(\mathbf{s}|y, x) = \mathcal{P}(\mathbf{s}|y) \Leftrightarrow \mathcal{P}(x|y, \mathbf{s}) = \mathcal{P}(x|y)$$

.

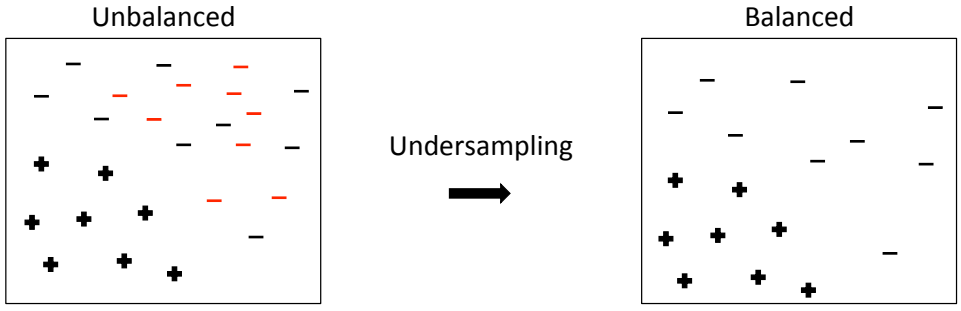
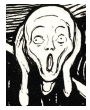


Figure : Undersampling: remove randomly majority class examples. In red samples that are removed from the unbalanced dataset ($\mathbf{s} = 0$).

Posterior probabilities

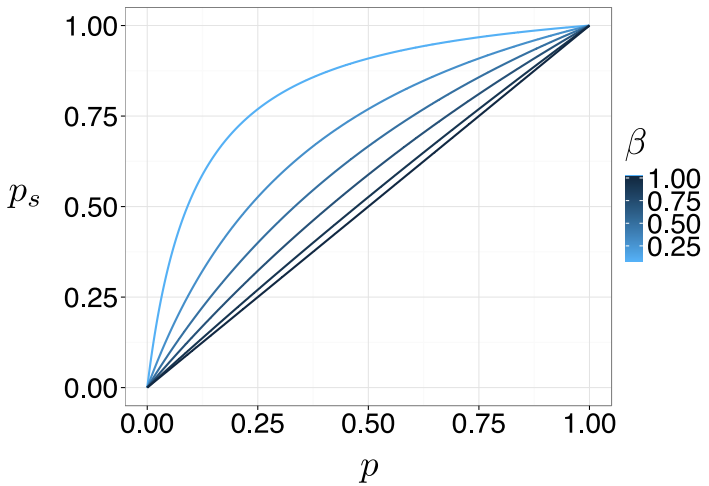


Let $p_s = p(+|x, \mathbf{s} = 1)$ and $p = p(+|x)$. We can write p_s as a function of p:

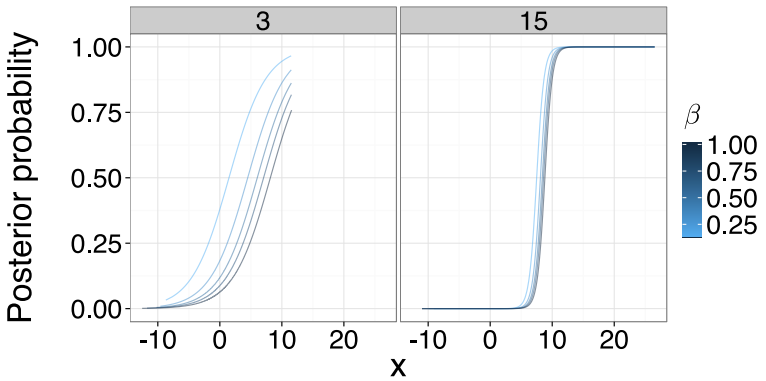
$$p_s = \frac{p}{p + \beta(1 - p)} \tag{1}$$

where $\beta = p(\mathbf{s} = 1|-)$. Using (1) we can obtain an expression of p as a function of p_s:

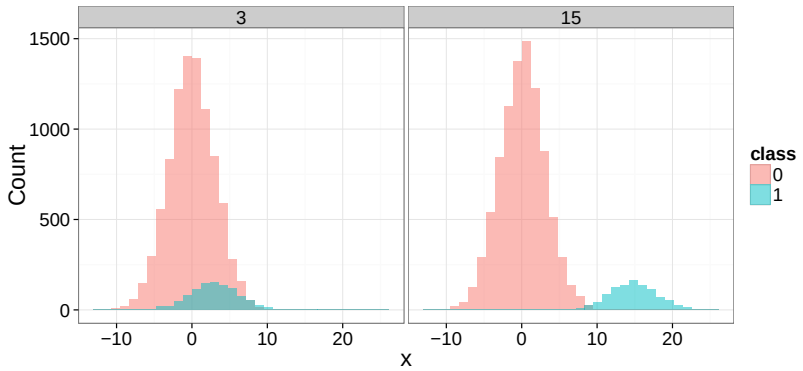
$$p = \frac{\beta p_s}{\beta p_s - p_s + 1} \tag{2}$$



Shift and class separability



(a) p_s as a function of β



(b) Class distribution

Figure : Class distribution and posterior probability as a function of β for two univariate binary classification tasks with norm class conditional densities $X^- \sim \mathcal{N}(0, \sigma)$ and $X^+ \sim \mathcal{N}(\mu, \sigma)$ (on the left $\mu = 3$ and on the right $\mu = 15$, in both examples $\sigma = 3$). Note that p corresponds to $\beta = 1$ and p_s to $\beta < 1$.

CONDITION FOR A BETTER RANKING

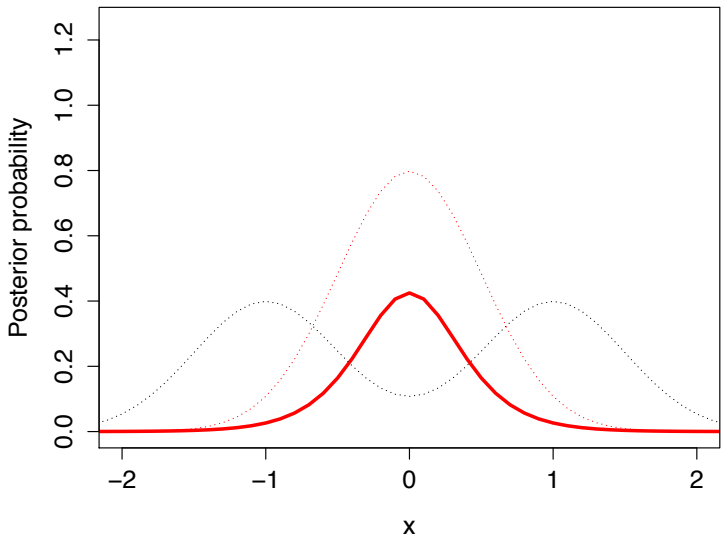
Using $\nu_s > \nu$ and by making a hypothesis of normality in the errors, $\mathcal{K}$ has better ranking with undersampling when

$$\boxed{\frac{dp_s}{dp} > \sqrt{\frac{\nu_s}{\nu}}} \tag{3}$$

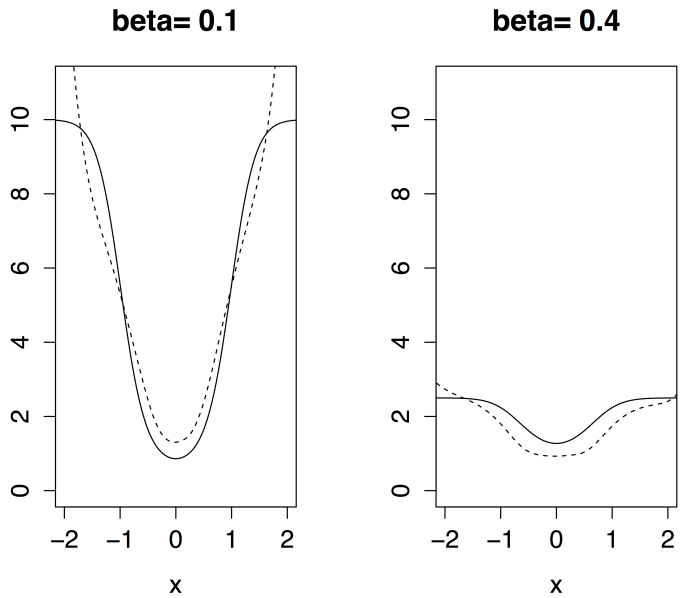
where $\frac{dp_s}{dp}$ is the derivative of p_s w.r.t. p:

$$\frac{dp_s}{dp} = \frac{\beta}{(p + \beta(1-p))^2}$$

UNIVARIATE SYNTHETIC DATASET



(a) Class conditional distributions (thin lines) and the posterior distribution of the minority class (thicker line).



(b) $\frac{dp_s}{dp}$ (solid lines), $\sqrt{\frac{\nu_s}{\nu}}$ (dotted lines).

Figure : Non separable case. On the right we plot both terms of inequality 3 (solid: left-hand, dotted: right-hand term) for $\beta = 0.1$ and $\beta = 0.4$

REAL DATASETS

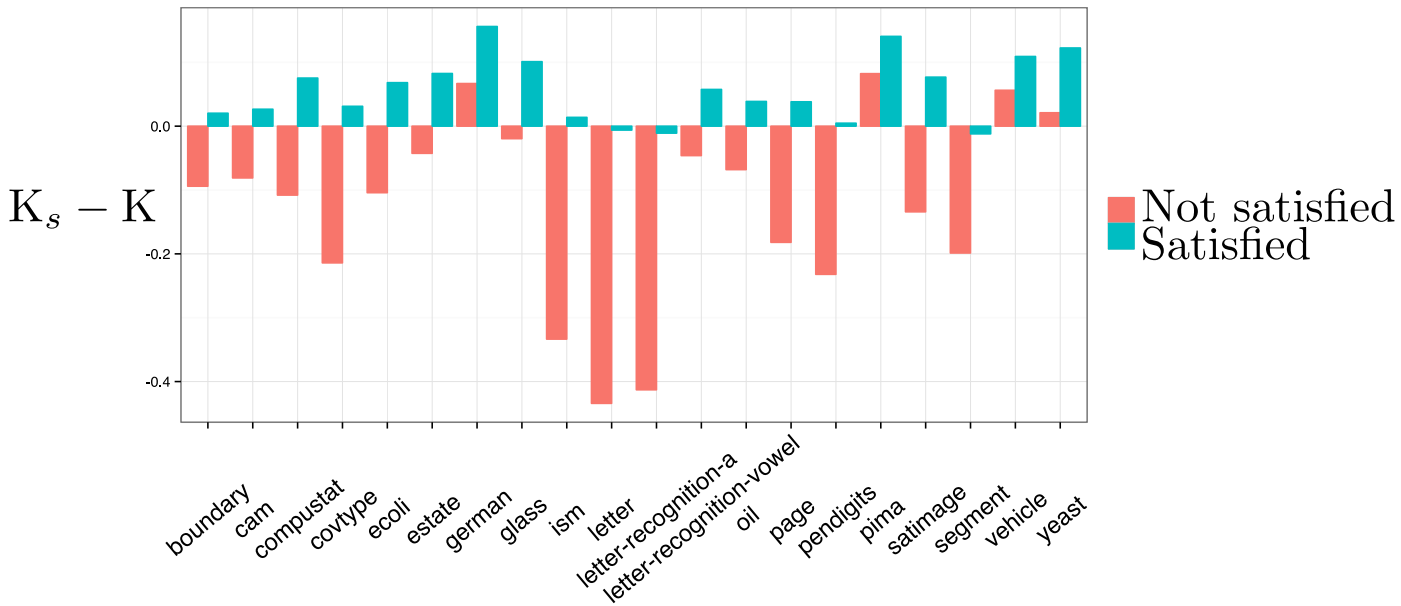


Figure : Difference between the Kendall rank correlation of $\hat{p}_s$ and $\hat{p}$ with p, namely K_s and K, for points having the condition (3) satisfied and not. K_s and K are calculated as the mean of the correlations over all βs.

Real datasets II

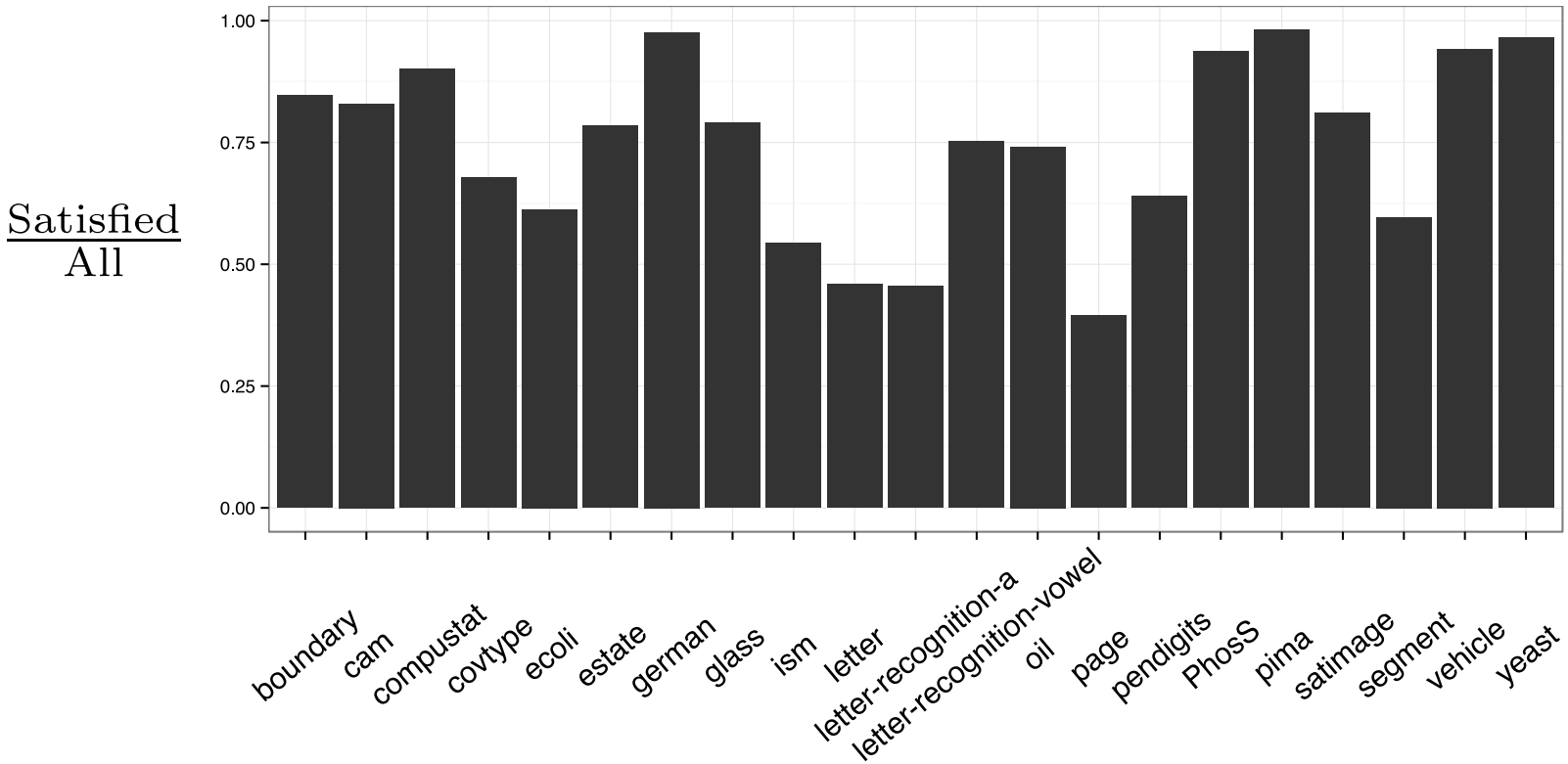


Figure : Ratio between the number of sample satisfying condition 3 and all the instances available in each dataset averaged over all the βs.

Discussion

- When (3) is satisfied the posterior probability obtained after sampling returns a more accurate ordering.
- Several factors influence (3) (e.g. β, variance of the classifier, class separability)
- Practical use (3) is not straightforward since it requires knowledge of p and $\frac{\nu_s}{\nu}$ (not easy to estimate).
- The sampling rate should be tuned, using a balanced distribution is not always the best option.

Selecting the best strategy

- With no prior information about the data distribution is difficult to decide which unbalanced strategy to use.
- *No-free-lunch theorem* [23]: no single strategy is coherently superior to all others in all conditions (i.e. algorithm, dataset and performance metric)
- Testing all unbalanced techniques is not an option because of the associated computational cost.
- We proposed to use the Racing approach [17] to perform strategy selection.

Racing for strategy selection

- Racing consists in testing in parallel a set of alternatives and using a statistical test to remove an alternative if it is significantly worse than the others.
- We adopted F-Race version [4] to search efficiently for the best strategy for unbalanced data.
- The F-race combines the Friedman test with Hoeffding Races [17].

RACING FOR UNBALANCED TECHNIQUE SELECTION

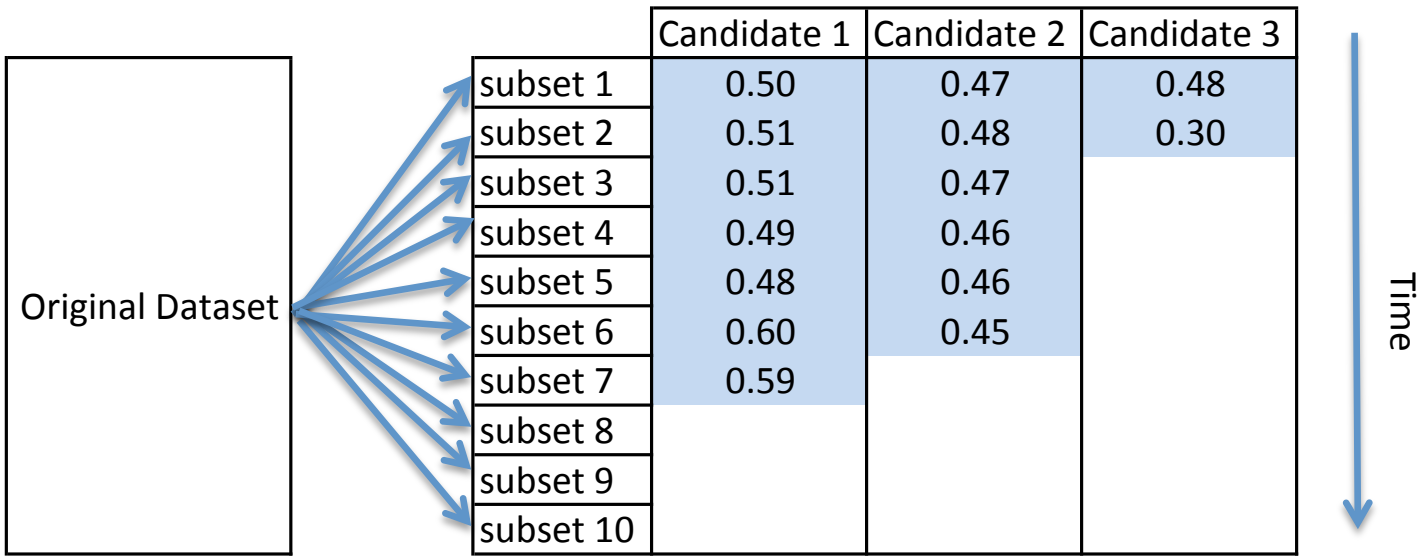
Automatically select the most adequate technique for a given dataset.

1. Test in parallel a set of alternative balancing strategies on a subset of the dataset
2. Remove progressively the alternatives which are significantly worse.
3. Iterate the testing and removal step until there is only one candidate left or not more data is available

	Candidate 1	Candidate 2	Candidate 3
subset 1	0.50	0.47	0.48
subset 2	0.51	0.48	0.30
subset 3	0.51	0.47	
subset 4	0.60	0.45	
subset 5	0.55		

F-RACE METHOD

- Use 10-fold Cross Validation (CV) to provide the data during the race.
- Every time new data is added to the race, the Friedman test is used to remove significantly bad candidates.
- We made a comparison of CV and F-race using different classifiers.



F-RACE VS CROSS VALIDATION

Dataset	Exploration	Method	Ntest	% Gain	Mean	Sd
ecoli	Race	Under	46	49	0.836	0.04
	CV	SMOTE	90	-	0.754	0.112
letter-a	Race	Under	34	62	0.952	0.008
	CV	SMOTE	90	-	0.949	0.01
letter-vowel	Race	Under	34	62	0.884	0.011
	CV	Under	90	-	0.887	0.009
letter	Race	SMOTE	37	59	0.951	0.009
	CV	Under	90	-	0.951	0.01
oil	Race	Under	41	54	0.629	0.074
	CV	SMOTE	90	-	0.597	0.076
page	Race	SMOTE	45	50	0.919	0.01
	CV	SMOTE	90	-	0.92	0.008
pendigits	Race	Under	39	57	0.978	0.011
	CV	Under	90	-	0.981	0.006
PhosS	Race	Under	19	79	0.598	0.01
	CV	Under	90	-	0.608	0.016
satimage	Race	Under	34	62	0.843	0.008
	CV	Under	90	-	0.841	0.011
segment	Race	SMOTE	90	0	0.978	0.01
	CV	SMOTE	90	-	0.978	0.01
estate	Race	Under	27	70	0.553	0.023
	CV	Under	90	-	0.563	0.021
covtype	Race	Under	42	53	0.924	0.007
	CV	SMOTE	90	-	0.921	0.008
cam	Race	Under	34	62	0.68	0.007
	CV	Under	90	-	0.674	0.015
compustat	Race	Under	37	59	0.738	0.021
	CV	Under	90	-	0.745	0.017
creditcard	Race	Under	43	52	0.927	0.008
	CV	SMOTE	90	-	0.924	0.006

Table : Results in terms of G-mean for RF classifier.

Discussion

- The best strategy is extremely dependent on the data nature, algorithm adopted and performance measure.
- F-race is able to automatise the selection of the best unbalanced strategy for a given unbalanced problem without exploring the whole dataset.
- For the fraud dataset the unbalanced strategy chosen had a big impact on the accuracy of the results.

Conclusions of Chapter 4

- Undersampling is not always the best strategy for all unbalanced datasets.
- This result warning against a naive use of undersampling in unbalanced tasks.
- With racing [9] we can rapidly select the best strategy for a given unbalanced task.
- However, we see that undersampling and SMOTE are often the best strategy to adopt for FD.

Methodological contribution

CHAPTER 5
-
LEARNING WITH CONCEPT DRIFT

Publications:

- A. Dal Pozzolo, O. Caelen, Y. Le Borgne, S. Waterschoot, and G. Bontempi.
  Learned lessons in credit card fraud detection from a practitioner perspective.
  Elsevier ESWA, 2014.
- A. Dal Pozzolo, R. Johnson, O. Caelen, S. Waterschoot, N. Chawla, and G. Bontempi.
  Using HDDT to avoid instances propagation in unbalanced and evolving data streams.
  In IEEE IJCNN, 2014.

Objective of Chapter 5

The objectives of this chapter are:

- investigating the benefit of updating the FDS.
- assessing the impact of Concept Drift (CD).
- comparing alternative learning strategies for CD.
- studying an effective way to deal with unbalanced distribution in a streaming environment.

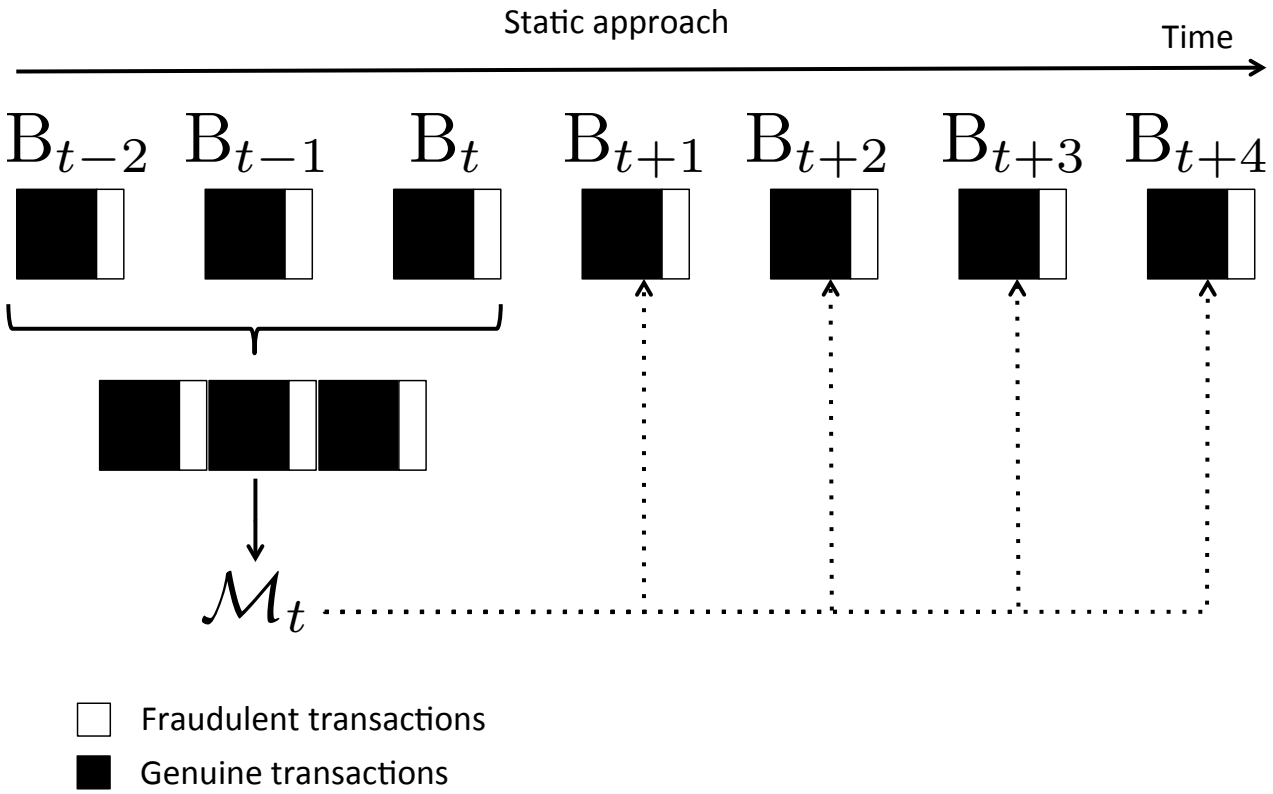
Learning approaches

We compare three standard learning approaches on transactions from February 2012 to May 2013.

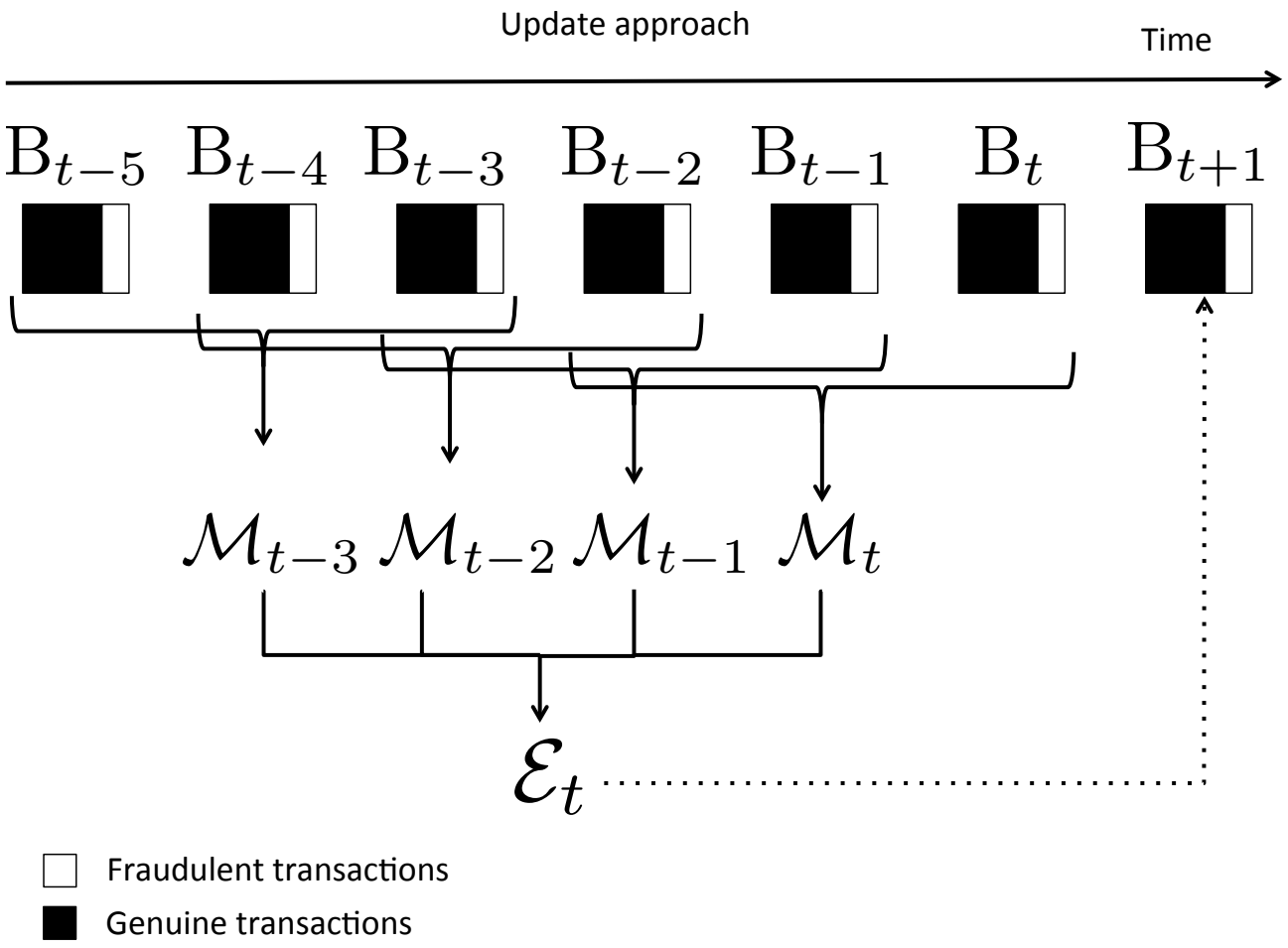
Approach	**Strengths**	**Weaknesses**
Static	• Speed	• No CD adaptation
Update	• No instances propagation • CD adaptation	• Need several batches for the minority class
Propagate and Forget	• Accumulates minority instances faster • CD adaptation	• Propagation leads to larger training time

# Days	# Features	# Transactions	Period
422	45	2'202'228	1Feb12 - 20May13

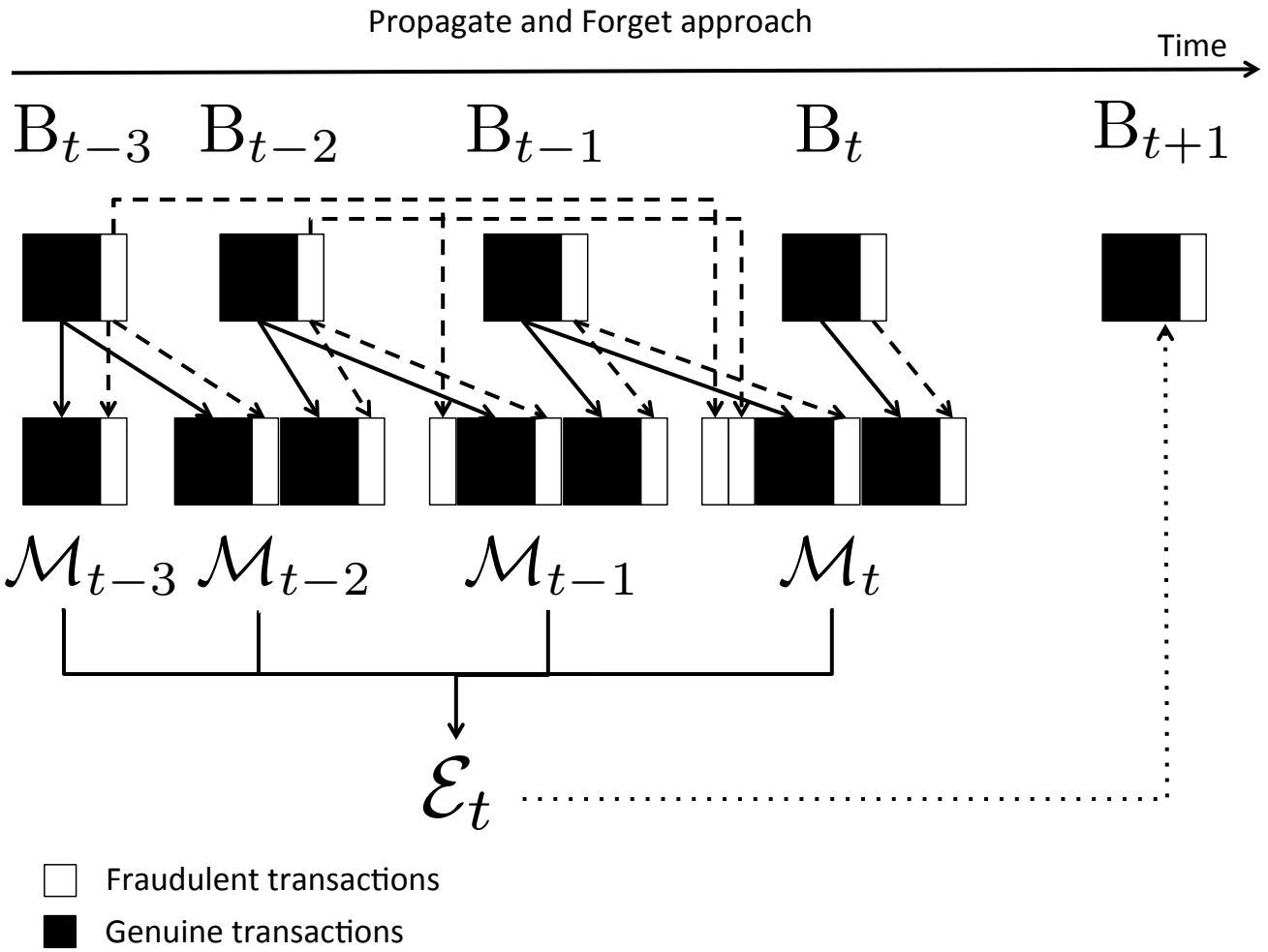
The Static approach



The Update approach



The Propagate and Forget approach



Best overall strategy

The *Propagate and Forget* approach is the best while the *Static* approach is the worst.

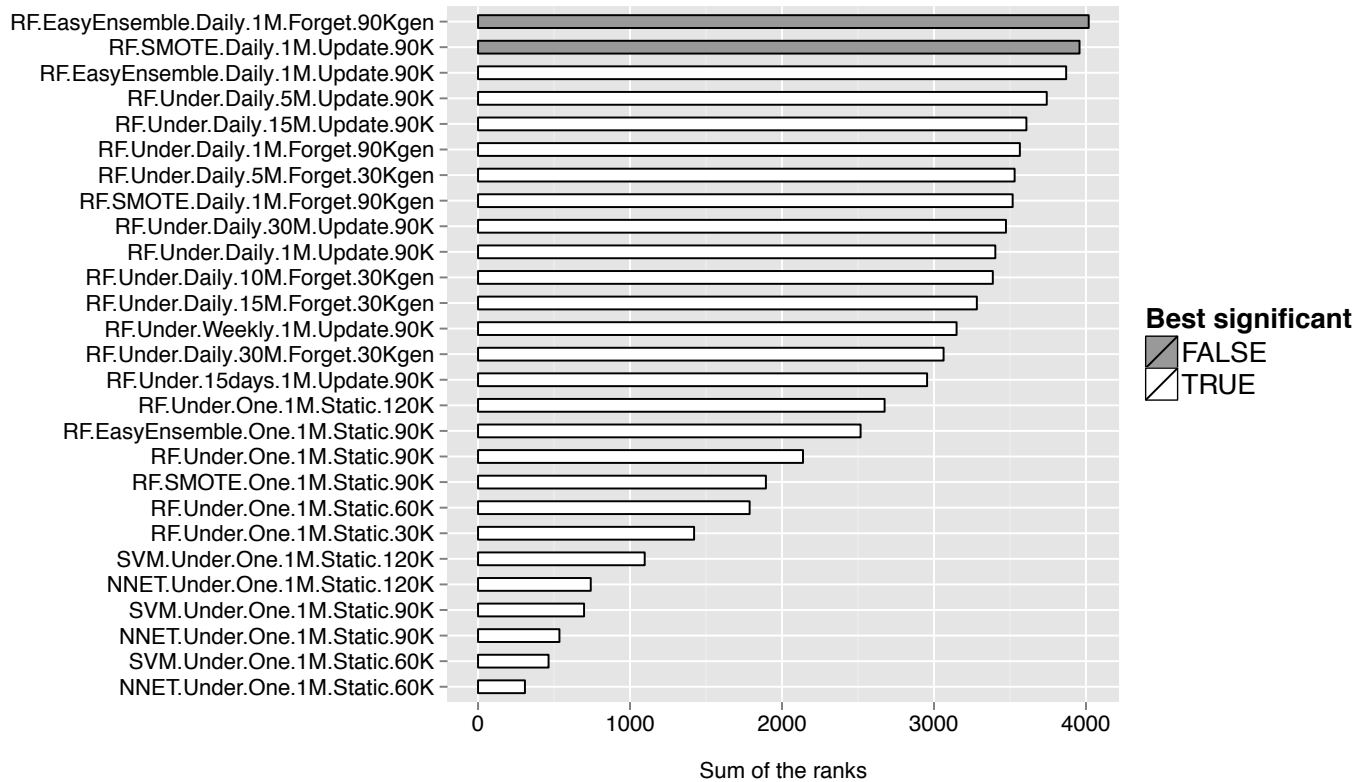


Figure : Comparison of all strategies using sum of ranks in all batches when using P_k as accuracy measure.

Discussion

- Without CD we would expect the three approaches to perform similarly.
- The static approach is consistently the worst.
- Ensembles are often better alternative than single models.
- Higher accuracy can be achieved by using SMOTE or undersampling the data.
- Instance propagation can be an alternative way to rebalance the data stream.

Conclusion of Chapter 5

- Updating the models daily and combining them into ensemble is an effective strategy to adapt to CD.
- Retaining positives samples/undersampling the data stream improves the accuracy of the FDS.
- However, it is not always possible to either revisit or store old instances of a data stream.
- In that case we reccomend using Hellinger Distance Decision Tree [7] (HDDT) to avoid instances propagation.

Proposed FDS

CHAPTER 6
-
LEARNING WITH ALERT-FEEDBACK INTERACTION

Publications:

- A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi.
  Credit Card Fraud Detection with Alert-Feedback Interaction.
  Submitted to IEEE TNNLS, 2015.
- A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi.
  Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information.
  In IEEE IJCNN, 2015.

Objective of Chapter 6

- Propose a framework meeting realistic working conditions.
- Model the interaction between the FDS generating alerts and investigators.
- Integrate investigators' feedback in the FDS.

Realistic working conditions

- Alert-Feedback Interaction: the supervised samples depends on the alerts reported to investigators.
- Only a small set of alerts can be checked by human investigators.
- The labels of the majority of transactions are available only several days later (after customers have reported unauthorized transactions).
- Investigators trust a FDS only if it provides accurate alerts, i.e. alert precision (P_k) is the accuracy measure that matters.

SUPERVISED SAMPLES

- The k transactions with largest $P_{\mathcal{K}_{t-1}}(+|x)$ define the alerts A_t reported to the investigators.
- Investigators provide a feedback about the alerts in A_t, defining a set of k supervised couples (x, y):

$$F_t = \{(x, y),\ x \in A_t\}, \tag{4}$$

- F_t are the only immediate supervised samples.
- $D_{t-\delta}$ are transactions that have not been checked by investigators, but their label is assumed to be correct after that δ days have elapsed.

SUPERVISED SAMPLES II

- F_t are a small set of risky transactions according the FDS.
- $D_{t-\delta}$ contains all the occurred transactions in a day ($\approx 99\%$ genuine transactions).

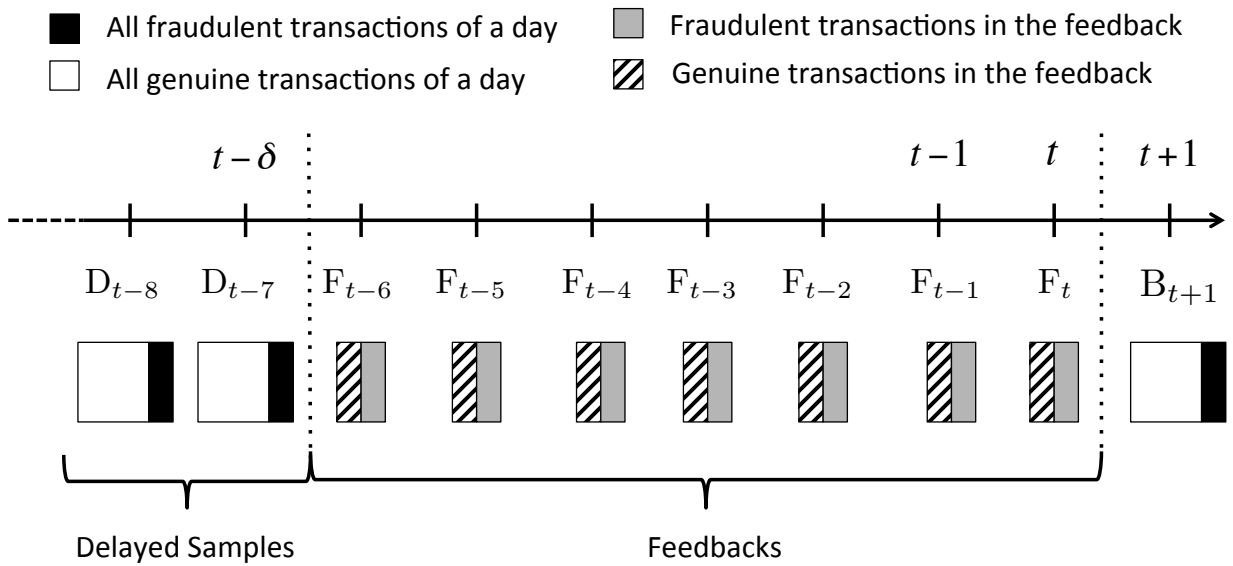


Figure : The supervised samples available at day t include: i) feedbacks of the first δ days and ii) delayed couples occurred before the δ^{th} day.

Feedbacks Vs Delayed samples

Learning from *feedbacks* F_t is a different problem than learning from *delayed samples* in $D_{t-\delta}$:

- F_t provides recent, up-to-date, information while $D_{t-\delta}$ might be already obsolete once it comes.
- Percentage of frauds in F_t and $D_{t-\delta}$ is different.
- Samples in F_t are selected as the most risky by $\mathcal{K}_{t-1}$.
- A classifier trained on F_t learns how to label transactions that are most likely to be fraudulent.

Feedbacks and delayed samples have to be treated separately.

Learning strategy

A conventional solution for CD adaptation is $\mathcal{W}_t$ [20]. To learn separately from feedbacks and delayed transactions we propose $\mathcal{A}_t^W$ (aggregation of $\mathcal{F}_t$ and $\mathcal{W}_t^D$).

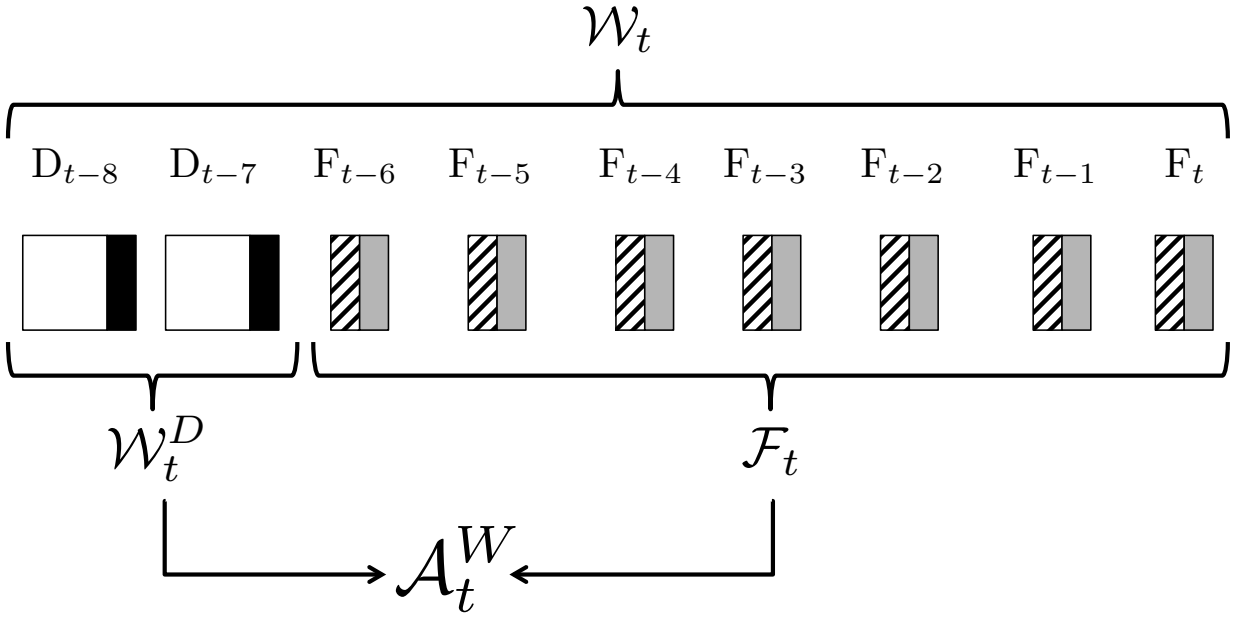


Figure : Learning strategies for feedbacks and delayed transactions occurring in the two days ($M = 2$) before the feedbacks ($\delta = 7$).

Classifier aggregations

$\mathcal{W}_t^D$ has to be aggregated with $\mathcal{F}_t$ to exploit information provided by feedbacks. We combine these classifiers by averaging the posterior probabilities.

$$\mathcal{P}_{\mathcal{A}_t^W}(+|x) = \alpha_t \mathcal{P}_{\mathcal{F}_t}(+|x) + (1 - \alpha_t)\mathcal{P}_{\mathcal{W}_t^D}(+|x) \tag{5}$$

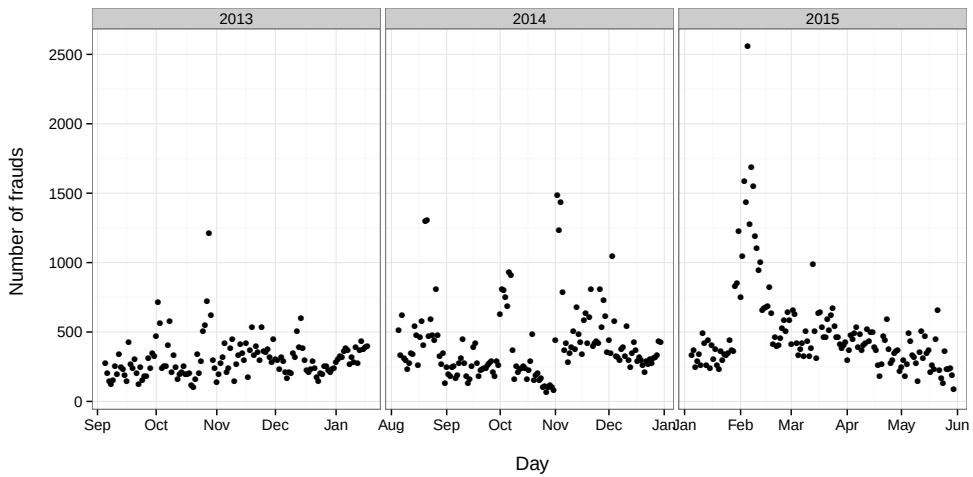
$\mathcal{A}_t^W$ with $\alpha_t \geq 0.5$ gives larger influence to feedbacks on the probability estimates w.r.t $\mathcal{W}_t$.

DATASETS

We used three datasets of e-commerce credit card transactions (fraud << 1%):

Table : Datasets

Id	Start day	End day	# Days	# Transactions	# Features
2013	2013-09-05	2014-01-18	136	21'830'330	51
2014	2014-08-05	2014-12-31	148	27'113'187	51
2015	2015-01-05	2015-05-31	146	27'651'197	51



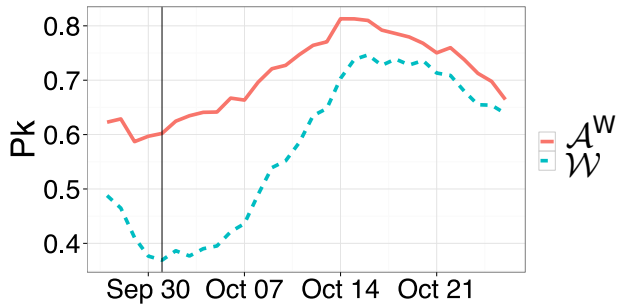
Separating feedbacks from delayed samples

Aggregating two distinct classifiers ($\mathcal{F}_t$ with $\mathcal{W}_t^D$) using $\mathcal{A}_t^W$ is better than pooling together feedbacks and delayed samples with $\mathcal{W}_t$.

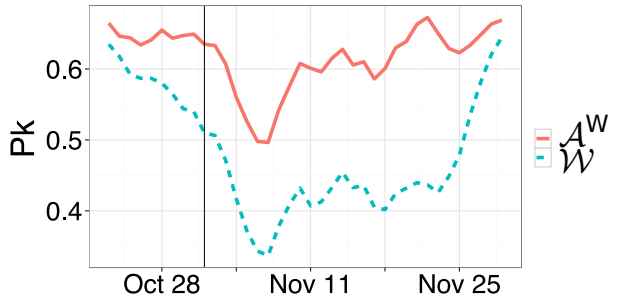
dataset	classifier	mean	sd
2013	$\mathcal{F}$	0.62	0.25
2013	$\mathcal{W}^D$	0.54	0.22
2013	$\mathcal{W}$	0.57	0.23
2013	$\mathcal{A}^W$	0.70	0.21
2014	$\mathcal{F}$	0.59	0.29
2014	$\mathcal{W}^D$	0.58	0.26
2014	$\mathcal{W}$	0.60	0.26
2014	$\mathcal{A}^W$	0.69	0.24
2015	$\mathcal{F}$	0.67	0.25
2015	$\mathcal{W}^D$	0.66	0.21
2015	$\mathcal{W}$	0.68	0.21
2015	$\mathcal{A}^W$	0.75	0.20

Table : Average P_k with $\delta = 7$, $M = 16$ and $\alpha_t = 0.5$.

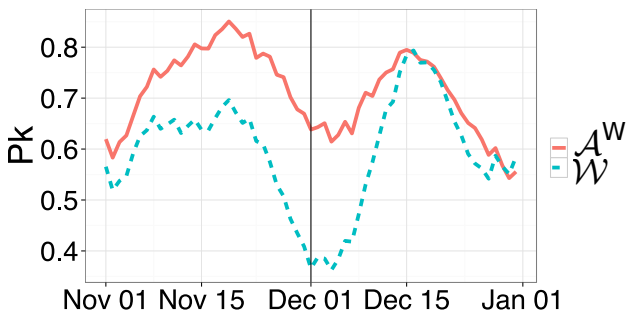
EXPERIMENTS WITH ARTIFICIAL CD



(a) **Sliding window** strategies on dataset **CD1**



(b) **Sliding window** strategies on dataset **CD2**



(c) **Sliding window** strategies on dataset **CD3**

Figure : Average P_k per day on datasets with artificial CD using moving average of 15 days. The vertical bar denotes the date of CD.

Classifiers ignoring AFI

Experiments with a classifier $\mathcal{R}_t$ trained on all recent transactions between t and $t-\delta$. $\mathcal{R}_t$ makes the unrealistic assumption that all transactions are labeled by investigators.

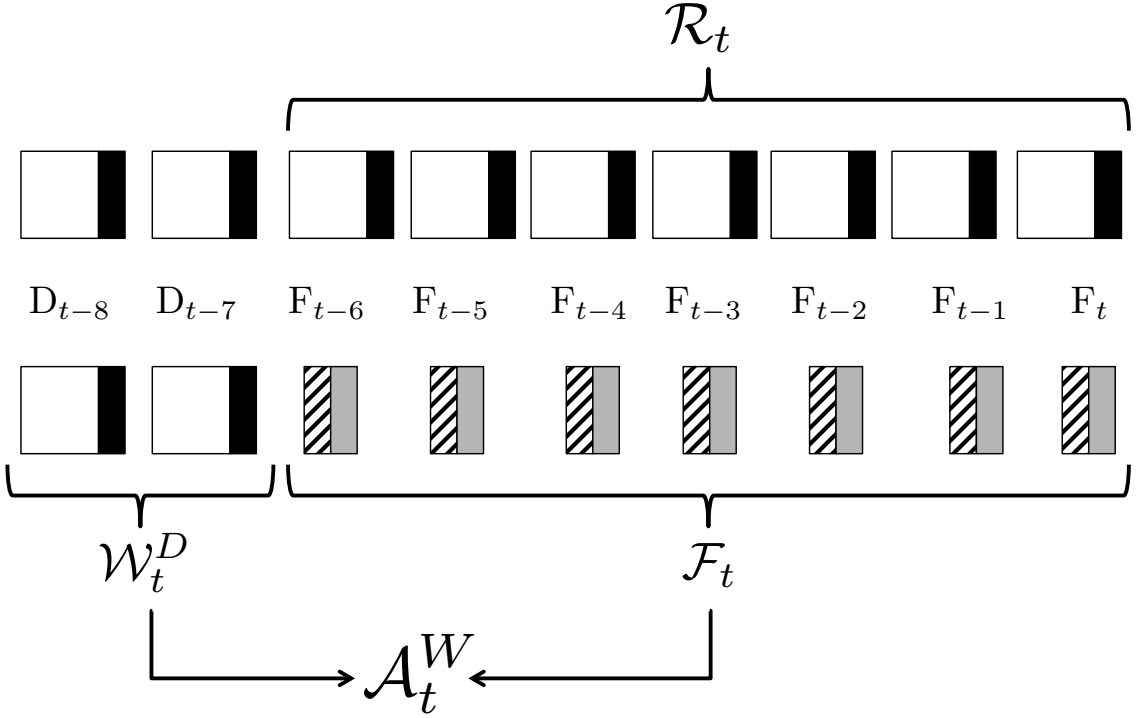


Figure : $\mathcal{R}_t$ makes the unrealistic assumption that all transactions between t and $t-\delta$ have been checked by investigators.

Classifiers ignoring AFI II

AUC is a global ranking measure, while P_k measures the ranking quality only on the top k.

(a) dataset **2013**

metric	classifier	mean	sd
AUC	$\mathcal{F}$	0.83	0.06
AUC	$\mathcal{W}^D$	0.94	0.03
AUC	$\mathcal{R}$	0.96	0.01
AUC	$\mathcal{A}^W$	0.94	0.03
P_k	$\mathcal{F}$	0.67	0.24
P_k	$\mathcal{W}^D$	0.54	0.22
P_k	$\mathcal{R}$	0.60	0.22
P_k	$\mathcal{A}^W$	0.72	0.20

(b) dataset **2014**

metric	classifier	mean	sd
AUC	$\mathcal{F}$	0.81	0.08
AUC	$\mathcal{W}^D$	0.94	0.03
AUC	$\mathcal{R}$	0.96	0.02
AUC	$\mathcal{A}^W$	0.93	0.03
P_k	$\mathcal{F}$	0.67	0.27
P_k	$\mathcal{W}^D$	0.56	0.27
P_k	$\mathcal{R}$	0.63	0.25
P_k	$\mathcal{A}^W$	0.71	0.25

(c) dataset **2015**

metric	classifier	mean	sd
AUC	$\mathcal{F}$	0.81	0.07
AUC	$\mathcal{W}^D$	0.95	0.02
AUC	$\mathcal{R}$	0.97	0.01
AUC	$\mathcal{A}^W$	0.94	0.02
P_k	$\mathcal{F}$	0.71	0.21
P_k	$\mathcal{W}^D$	0.62	0.21
P_k	$\mathcal{R}$	0.68	0.20
P_k	$\mathcal{A}^W$	0.76	0.19

Table : Average AUC and P_k for the sliding approach ($\delta = 15$, $M = 16$ and $\alpha_t = 0.5$).

Adaptive weighting

Experiments with different weighting strategies in $\mathcal{A}_t^W$.

(a) dataset **2013**

α adaptation	mean	sd
probDif	0.70	0.22
acc	0.72	0.21
accMiss	0.72	0.21
precision	0.72	0.21
recall	0.71	0.21
auc	0.72	0.21
none($\alpha_t = 0.5$)	0.72	0.20

(b) dataset **2014**

α adaptation	mean	sd
probDif	0.68	0.26
acc	0.71	0.24
accMiss	0.70	0.24
precision	0.71	0.24
recall	0.70	0.25
auc	0.71	0.24
none($\alpha_t = 0.5$)	0.71	0.24

(c) dataset **2015**

α adaptation	mean	sd
probDif	0.73	0.21
acc	0.76	0.19
accMiss	0.75	0.19
precision	0.75	0.19
recall	0.74	0.20
auc	0.75	0.19
none($\alpha_t = 0.5$)	0.76	0.19

Table : Average P_k of $\mathcal{A}_t^W$ with adaptive α_t ($\delta = 15$).

Adaptive weighting II

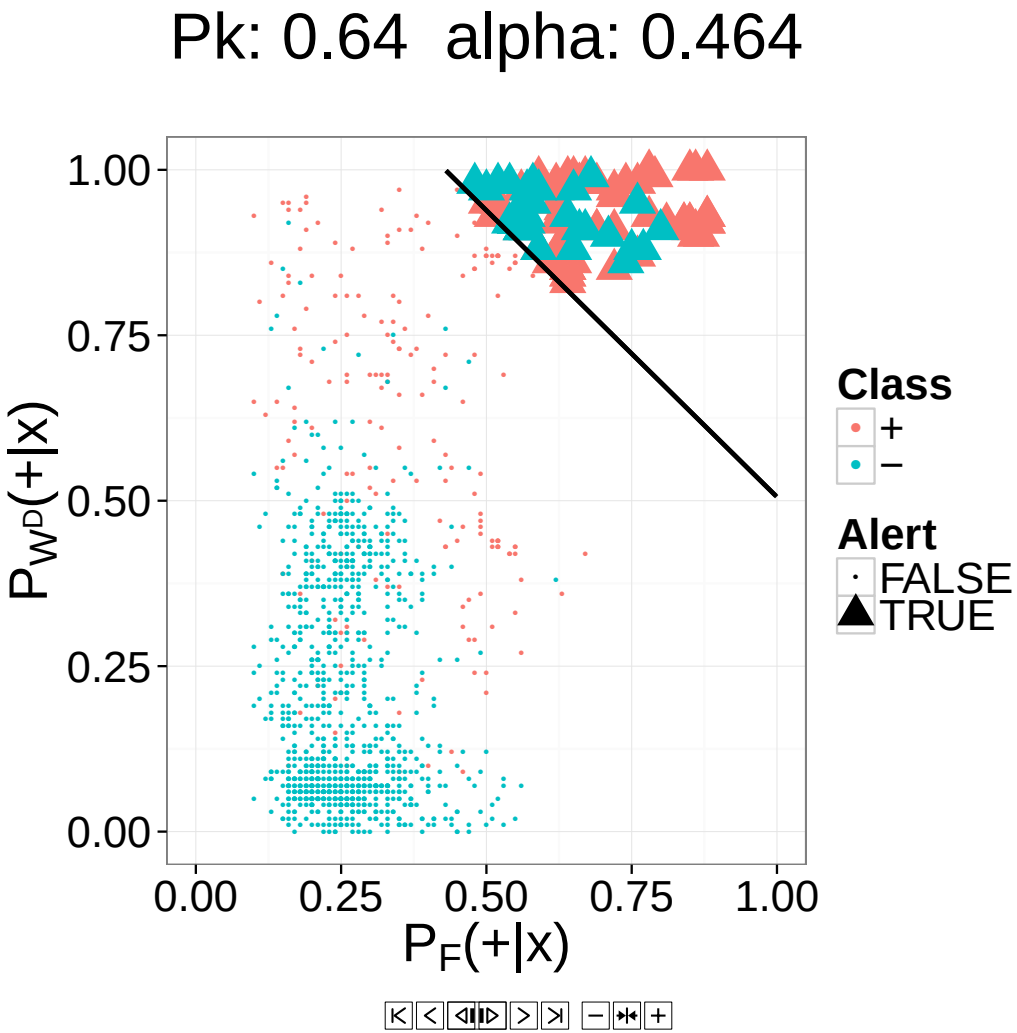


Figure : $\mathcal{P}_{\mathcal{F}_t}(+|x)$ and $\mathcal{P}_{\mathcal{W}_t^D}(+|x)$ for alerts in different days.

Conclusion of Chapter 6

- The goal of the FDS is to return precise alerts (high P_k).
- In a realistic settings feedbacks and delayed samples are the only supervised information available.
- Feedbacks and delayed samples address two different classification tasks.
- Not convenient to pool the two types of supervised samples together.

CONCLUSIONS AND FUTURE PERSPECTIVES

Summary of contributions

In this thesis we:

- formalize realistic working conditions of a FDS.
- study how undersampling can improve the ranking of a classifier $\mathcal{K}$.
- release a R package called `unbalanced` [8] with racing.
- investigate strategies for data streams with both CD and unbalanced distribution.
- prototype a real-world FDS that, for the first time, learns from investigators' feedback and allows us to have more precise alerts.

Learned lessons

- Rebalancing a training set with undersampling is not guaranteed to improve performances.
- F-Race [4] can be effective to rapidly select the unbalanced technique to use.
- The data stream defined by credit card transactions is severely affected by CD, updating the FDS is often a good idea.
- For a real-world FDS it is mandatory to produce precise alerts.
- Feedbacks and delayed samples should be separately handled when training a FDS.
- Aggregating two distinct classifiers is an effective strategy and that it enables a quicker adaptation to CD.

Take-home messages

- Class imbalance and CD seriously affect the performances of a classifier.
- In a real-world scenario, there is a strong Alert-Feedback interaction that has to be explicitly considered.
- The best learning algorithm for a FDS changes with the accuracy measure.
- A standard classifier that is trained on all recent transactions is not the best when we want to produce accurate alerts.

Future work: towards Big Data solutions

- This work will be continued in the *BruFence* project.
- Big Data Mining for Fraud Detection and Security.
- Three research groups (ULB-QualSec, ULB-MLG and UCL-MLG) and three companies (Worldline, Steria and Nviso).
- 2015 - 2018 funded by InnovIRIS (Brussels Region).

THE *BruFence* PROJECT

- Goal: implement the FDS of Chapter 6 using Big Data technologies (e.g. Spark, NoSQL DB).
- Real-time framework scalable w.r.t. the amount of data and resources available.
- Determine to which extent external network data can improve prediction accuracy.
- Compute aggregated features in real time and using a larger set of historical transactions.

Figure : Apache Big Data softwares

Added value for WL

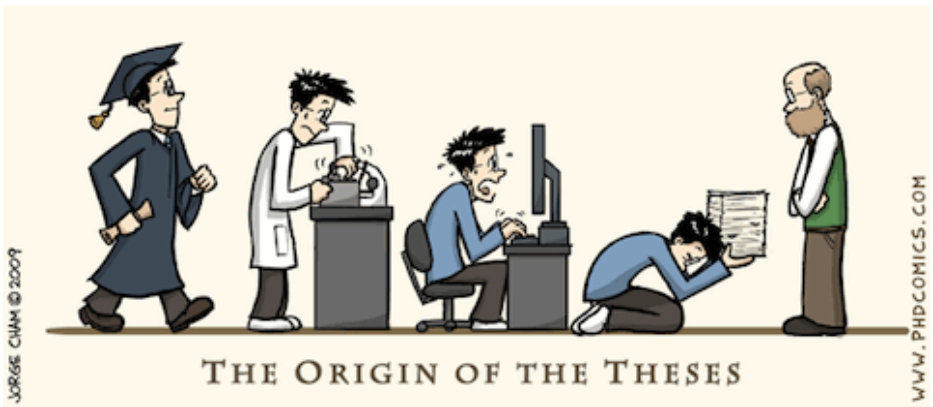
- RF has emerged as the best algorithm (now used by WL).
- New ways to create aggregated features.
- Undersampling can be used to improve the accuracy of FDS, but probabilities need to be calibrated.
- Our experimental analysis on CD gave WL some guidelines on how to improve the FDS.
- Including feedbacks in the learning process allows WL to obtain more accurate alerts.

Concluding remarks

- *Doctiris* was an unique opportunity to work on real-world fraud detection data.
- Real working conditions define new challenges (often ignored in the literature).
- Companies are more interested on practical results, while academics on theoretical ones.
- Industry and university, should look at each other and exchange ideas in order to have a much larger impact on society.

Website: `www.ulb.ac.be/di/map/adalpozz`
Email: adalpozz@ulb.ac.be

Thank you for the attention



Bibliography I

[1] D. N. A. Asuncion.
UCI machine learning repository, 2007.

[2] C. Alippi, G. Boracchi, and M. Roveri.
A just-in-time adaptive classification system based on the intersection of confidence intervals rule.
Neural Networks, 24(8):791–800, 2011.

[3] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer.
Moa: Massive online analysis.
The Journal of Machine Learning Research, 99:1601–1604, 2010.

[4] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp.
A racing algorithm for configuring metaheuristics.
In *Proceedings of the genetic and evolutionary computation conference*, pages 11–18, 2002.

[5] D. A. Cieslak and N. V. Chawla.
Detecting fractures in classifier performance.
In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 123–132. IEEE, 2007.

[6] D. A. Cieslak and N. V. Chawla.
Learning decision trees for unbalanced data.
In *Machine Learning and Knowledge Discovery in Databases*, pages 241–256. Springer, 2008.

[7] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer.
Hellinger distance decision trees are robust and skew-insensitive.
Data Mining and Knowledge Discovery, 24(1):136–158, 2012.

[8] A. Dal Pozzolo, O. Caelen, and G. Bontempi.
unbalanced: Racing For Unbalanced Methods Selection., 2015.
R package version 2.0.

Bibliography II

[9] A. Dal Pozzolo, O. Caelen, S. Waterschoot, and G. Bontempi.
Racing for unbalanced methods selection.
In *Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning*. IDEAL, 2013.

[10] C. Elkan.
The foundations of cost-sensitive learning.
In *International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978. Citeseer, 2001.

[11] R. Elwell and R. Polikar.
Incremental learning of concept drift in nonstationary environments.
Neural Networks, IEEE Transactions on, 22(10):1517–1531, 2011.

[12] J. Gao, B. Ding, W. Fan, J. Han, and P. S. Yu.
Classifying data streams with skewed class distributions and concept drifts.
Internet Computing, 12(6):37–49, 2008.

[13] T. R. Hoens, R. Polikar, and N. V. Chawla.
Learning from streaming data with concept drift and imbalance: an overview.
Progress in Artificial Intelligence, 1(1):89–101, 2012.

[14] M. G. Kelly, D. J. Hand, and N. M. Adams.
The impact of changing populations on classifier performance.
In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 367–371. ACM, 1999.

[15] A. Liaw and M. Wiener.
Classification and regression by randomforest.
R News, 2(3):18–22, 2002.

Bibliography III

[16] R. N. Lichtenwalter and N. V. Chawla.
Adaptive methods for classification in arbitrarily imbalanced and drifting data streams.
In *New Frontiers in Applied Data Mining*, pages 53–75. Springer, 2010.

[17] O. Maron and A. Moore.
Hoeffding races: Accelerating model selection search for classification and function approximation.
Robotics Institute, page 263, 1993.

[18] C. R. Rao.
A review of canonical coordinates and an alternative to correspondence analysis using hellinger distance.
Questiió: Quaderns d'Estadística, Sistemes, Informatica i Investigació Operativa, 19(1):23–63, 1995.

[19] M. Saerens, P. Latinne, and C. Decaestecker.
Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure.
Neural computation, 14(1):21–41, 2002.

[20] D. K. Tasoulis, N. M. Adams, and D. J. Hand.
Unsupervised clustering in streaming data.
In *ICDM Workshops*, pages 638–642, 2006.

[21] V. N. Vapnik and V. Vapnik.
Statistical learning theory, volume 1.
Wiley New York, 1998.

[22] S. Wang, K. Tang, and X. Yao.
Diversity exploration and negative correlation learning on imbalanced data sets.
In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 3259–3266. IEEE, 2009.

[23] D. H. Wolpert.
The lack of a priori distinctions between learning algorithms.
Neural computation, 8(7):1341–1390, 1996.

Bibliography IV

Open issues

- Agreeing on a good performance measure (cost-based Vs detection).
- Modelling Alert-Feedback Interaction.
- Using both the supervised and unsupervised information available.

Average Precision

Let N^+ be the number of positive (fraud) case in the original dataset and TP_k be the number of true positives in the first k ranked transactions ($TP_k \leq k$). Let us denote Precision and Recall at k as $\mathrm{P}_k = \frac{TP_k}{k}$ and $\mathrm{R}_k = \frac{TP_k}{N^+}$. We can then define AP as:

$$\mathrm{AP} = \sum_{k=1}^{N} \mathrm{P}_k(\mathrm{R}_k - \mathrm{R}_{k-1}) \tag{6}$$

where N is the total number of observation in the dataset.

Ranking error

We have a wrong ranking if $\hat{p}_1 > \hat{p}_2$ and its probability is:

$$\mathcal{P}(\hat{p}_2 < \hat{p}_1) = \mathcal{P}(p_2 + \epsilon_2 < p_1 + \epsilon_1) = \mathcal{P}(\epsilon_1 - \epsilon_2 > \Delta p)$$

where $\epsilon_2 - \epsilon_1 \sim \mathcal{N}(0, 2\nu)$. By making an hypothesis of normality we have

$$\mathcal{P}(\epsilon_1 - \epsilon_2 > \Delta p) = 1 - \Phi\left(\frac{\Delta p}{\sqrt{2\nu}}\right) \tag{7}$$

where Φ is the cumulative distribution function of the standard normal distribution.
The probability of a ranking error with undersampling is:

$$\mathcal{P}(\hat{p}_{s,2} < \hat{p}_{s,1}) = \mathcal{P}(\eta_1 - \eta_2 > \Delta p_s)$$

and

$$\mathcal{P}(\eta_1 - \eta_2 > \Delta p_s) = 1 - \Phi\left(\frac{\Delta p_s}{\sqrt{2\nu_s}}\right) \tag{8}$$

Classification threshold

Let r^+ and r^- be the risk of predicting an instance as positive and negative:

$$r^+ = (1-p) \cdot l_{1,0} + p \cdot l_{1,1}$$

$$r^- = (1-p) \cdot l_{0,0} + p \cdot l_{0,1}$$

where $l_{i,j}$ is the cost in predicting i when the true class is j and $p = \mathcal{P}(y = +|x)$. A sample is predicted as positive if $r^+ \leq r^-$ [21].

Alternatively, predict as positive when $p > \tau$ with τ:

$$\tau = \frac{l_{1,0} - l_{0,0}}{l_{1,0} - l_{0,0} + l_{0,1} - l_{1,1}} \tag{9}$$

CONDITION FOR A BETTER RANKING

$\mathcal{K}$ has better ranking with undersampling w.r.t. a classifier learned with unbalanced distribution when

$$\mathcal{P}(\hat{p}_2 < \hat{p}_1) > \mathcal{P}(\hat{p}_{s,2} < \hat{p}_{s,1})$$

$$\mathcal{P}(\epsilon_1 - \epsilon_2 > \Delta p) > \mathcal{P}(\eta_1 - \eta_2 > \Delta p_s) \tag{10}$$

or equivalently from (7) and (8) when

$$1 - \Phi\left(\frac{\Delta p}{\sqrt{2\nu}}\right) > 1 - \Phi\left(\frac{\Delta p_s}{\sqrt{2\nu_s}}\right) \Leftrightarrow \Phi\left(\frac{\Delta p}{\sqrt{2\nu}}\right) < \Phi\left(\frac{\Delta p_s}{\sqrt{2\nu_s}}\right)$$

since Φ is monotone non decreasing and $\nu_s > \nu$: Then it follows that undersampling is useful (better ranking) when

$$\frac{dp_s}{dp} > \sqrt{\frac{\nu_s}{\nu}}$$

Correcting posterior probabilities

We propose to use p', the bias-corrected probability obtained from p_s:

$$p' = \frac{\beta p_s}{\beta p_s - p_s + 1} \tag{11}$$

Eq. (11) is a special case of the framework proposed by Saerens et al. [19] and Elkan [10] for correcting the posterior probability in the case of testing and training sets sharing the same priors.

CORRECTING THE CLASSIFICATION THRESHOLD

When the costs of a FN ($l_{0,1}$) and FP ($l_{1,0}$) are unknown, we can use the priors. Let $l_{1,0} = \pi^+$ and $l_{0,1} = \pi^-$, from (9) we get:

$$\tau = \frac{l_{1,0}}{l_{1,0} + l_{0,1}} = \frac{\pi^+}{\pi^+ + \pi^-} = \pi^+ \tag{12}$$

since $\pi^+ + \pi^- = 1$. Then we should use π^+ as threshold with p:

$$p \longrightarrow \tau = \pi^+$$

Similarly

$$p_s \longrightarrow \tau_s = \pi_s^+$$

From Elkan [10]:

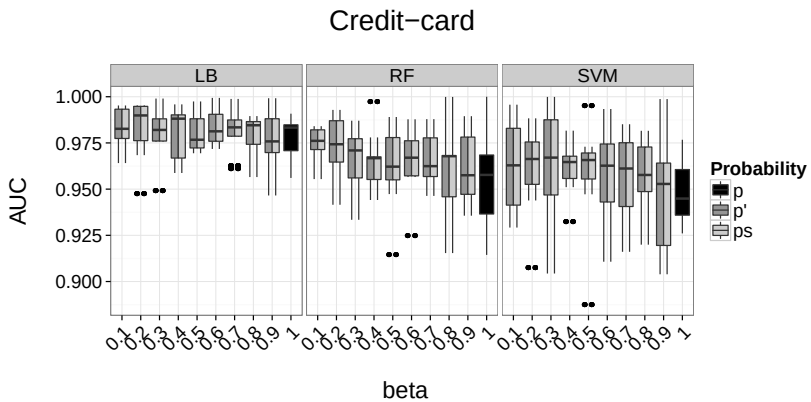
$$\frac{\tau'}{1-\tau'} \frac{1-\tau_s}{\tau_s} = \beta \tag{13}$$

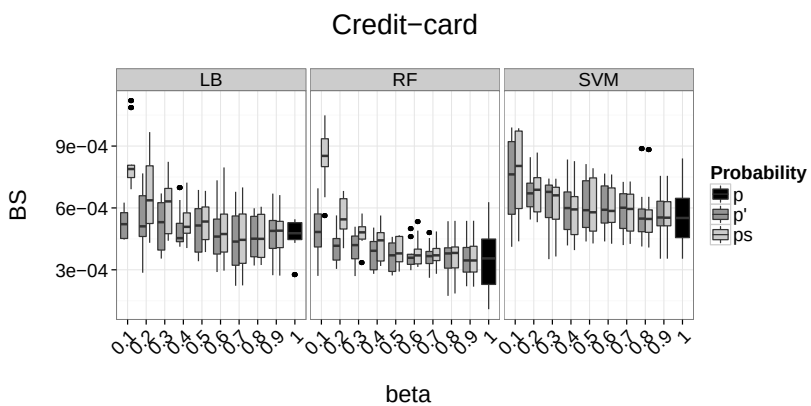
Therefore, we obtain:

$$p' \longrightarrow \tau' = \pi^+$$

CREDIT CARDS DATASET

Real-world credit card dataset with transactions from Sep 2013, frauds account for 0.172% of all transactions.





Discussion

- As a result of undersampling, $\hat{p}_s$ is shifted away from $\hat{p}$.
- Using (11), we can remove the drift in $\hat{p}_s$ and obtain $\hat{p}'$ which has better calibration.
- $\hat{p}'$ provides the same ranking quality of $\hat{p}_s$.
- Using $\hat{p}'$ with τ' we are able to improve calibration without losing predictive accuracy.
- Unfortunately there is no way to select the best β automatically yet.

Hellinger Distance

- Quantify the similarity between two probability distributions [18]
- Recently proposed as a splitting criteria in decision trees for unbalanced problems [6, 7].
- In data streams, it has been used to detect classifier performance degradation due to concept drift [5, 16].

Hellinger Distance II

Consider two probability measures P_1, P_2 of discrete distributions in Φ. The Hellinger distance is defined as:

$$HD(P_1, P_2) = \sqrt{\sum_{\phi \in \Phi} \left(\sqrt{P_1(\phi)} - \sqrt{P_2(\phi)} \right)^2} \tag{14}$$

Hellinger distance has several properties:

- $HD(P_1, P_2) = HD(P_1, P_2)$ (symmetric)
- $HD(P_1, P_2) >= 0$ (non-negative)
- $HD(P_1, P_2) \in [0, \sqrt{2}]$

HD is close to zero when the distributions are similar and close to $\sqrt{2}$ for distinct distributions.

Hellinger Distance Decision Trees

- HDDT [6] use Hellinger Distance as splitting criteria in decision trees for unbalanced problems.
- All numerical features are partitioned into q bins (only categorical variables).
- The Hellinger Distance between the positive and negative class is computed for each feature and then the feature with the maximum distance is used to split the tree.

$$HD(f^+, f^-) = \sqrt{\sum_{j=1}^{q} \left(\sqrt{\frac{|f_j^+|}{|f^+|}} - \sqrt{\frac{|f_j^-|}{|f^-|}} \right)^2} \tag{15}$$

where j defines the j^{th} bin of feature f and $|f_j^+|$ is the number of positive instances of feature f in the j^{th} bin.
Note that the class **priors do not appear** explicitly in (15).

Avoiding instances propagation

- HDDT allows us to remove instance propagations between batches, benefits:
  - improved predictive accuracy
  - improved speed
  - single-pass through the data.
- An ensemble of HDDTs is used to adapt to CD and increase the accuracy of single classifiers.
- We test our framework on several streaming datasets with unbalanced classes and concept drift.

Datasets

We used different types of datasets.

- UCI datasets [1] (unbalanced, no CD).
- Synthetic datasets from MOA [3] (artificial CD).
- A credit card dataset (online payment transactions between the 5^{th} and the 25^{th} of Sept. 2013).

Name	Source	Instances	Features	Imbalance Ratio
Adult	UCI	48,842	14	3.2:1
can	UCI	443,872	9	52.1:1
compustat	UCI	13,657	20	27.5:1
covtype	UCI	38,500	10	13.0:1
football	UCI	4,288	13	1.7:1
ozone-8h	UCI	2,534	72	14.8:1
wrds	UCI	99,200	41	1.0:1
text	UCI	11,162	11465	14.7:1
DriftedLED	MOA	1,000,000	25	9.0:1
DriftedRBF	MOA	1,000,000	11	1.02:1
DriftedWave	MOA	1,000,000	41	2.02:1
Creditcard	FRAUD	3,143,423	36	658.8:1

Results Credit card dataset

Figure 18 shows the sum of the ranks for each strategy over all the batches. For each batch, we assign the highest rank to the most accurate strategy and then sum the ranks over all batches.

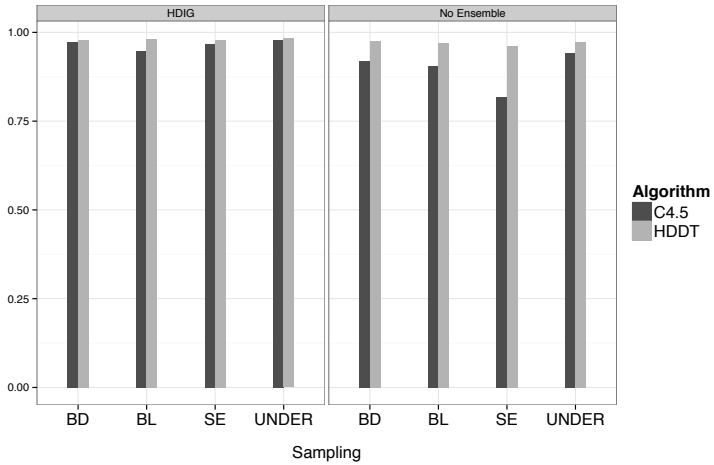


Figure : Batch average results in terms of **AUROC** (higher is better) using different sampling strategies and batch-ensemble weighting methods with C4.5 and HDDT over **Credit card dataset**.

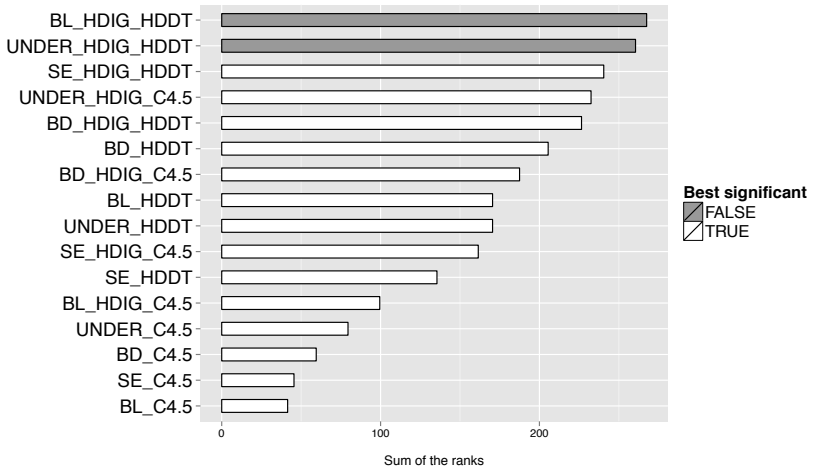


Figure : Sum of ranks in all chunks for the **Credit card dataset** in terms of **AUROC**. In gray are the strategies that are not significantly worse than the best having the highest sum of ranks.

Hellinger distance between batches

Let B_t be the batch at time t used for training $\mathcal{K}_t$ and B_{t+1} the testing batch. We compute the distance between B_t and B_{t+1} for a given feature f using Hellinger distance as [16]:

$$HD(B_t^f, B_{t+1}^f) = \sqrt{\sum_{v \in f} \left(\sqrt{\frac{|B_t^{f=v}|}{|B_t|}} - \sqrt{\frac{|B_{t+1}^{f=v}|}{|B_{t+1}|}} \right)^2} \tag{16}$$

where $|B_t^{f=v}|$ is the number of instances of feature f taking value v in the batch at time t, while $|B_t|$ is the total number of instances in the same batch.

Hellinger distance and Information Gain

We can make the assumption that feature relevance remains stable over time and define a new distance function that combines IG and HD as:

$$HDIG(\mathrm{B}_t, \mathrm{B}_{t+1}, f) = HD(\mathrm{B}_t^f, \mathrm{B}_{t+1}^f) * (1 + IG(\mathrm{B}_t, f)) \tag{17}$$

The final distance between two batches is then computed taking the average over all the features.

$$AHDIG(\mathrm{B}_t, \mathrm{B}_{t+1}) = \frac{\sum_{f \in \mathrm{B}_t} HDIG(\mathrm{B}_t, \mathrm{B}_{t+1}, f)}{|f \in \mathrm{B}_t|} \tag{18}$$

Then $\mathcal{K}_t$ receives as weight $w_t = AHDIG(\mathrm{B}_t, \mathrm{B}_{t+1})^{-M}$, where M is the ensemble size.

Adaptive weighting II

Let $F_{\mathcal{A}_t^W}$ be the feedbacks requested by $\mathcal{A}_t^W$ and $F_{\mathcal{F}_t}$ be the feedbacks requested by $\mathcal{F}_t$. Let Y_t^+ (resp. Y_t^-) be the fraudulent (resp. genuine) transactions at day t (B_t), we define the following sets:

- $CORR_{\mathcal{F}_t} = F_{\mathcal{A}_t^W} \cap F_{\mathcal{F}_t} \cap Y_t^+$.
- $ERR_{\mathcal{F}_t} = F_{\mathcal{A}_t^W} \cap F_{\mathcal{F}_t} \cap Y_t^-$.
- $MISS_{\mathcal{F}_t} = F_{\mathcal{A}_t^W} \cap \{B_t \setminus F_{\mathcal{F}_t}\} \cap Y_t^+$.

ADAPTIVE WEIGHTING II

In the example below we have $|\text{CORR}_{\mathcal{F}_t}| = 4$, $|\text{ERR}_{\mathcal{F}_t}| = 2$ and $|\text{MISS}_{\mathcal{F}_t}| = 3$.

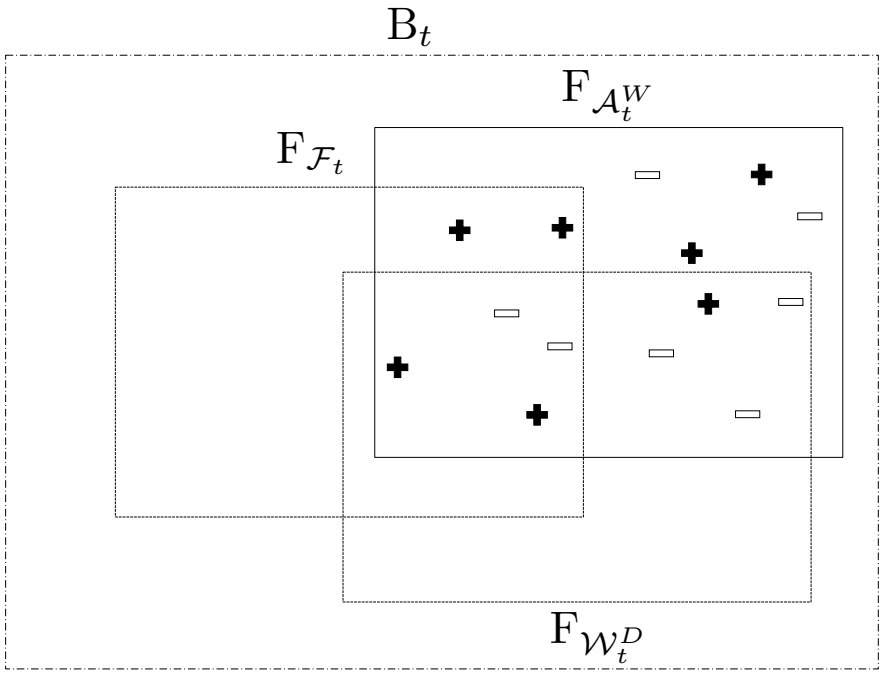


Figure : Feedbacks requested by $\mathcal{A}_t^W$ ($F_{\mathcal{A}_t^W}$) are a subset of all the transactions of day t (B_t). $F_{\mathcal{F}_t}$ ($F_{\mathcal{W}_t^D}$) denotes the feedbacks requested by $\mathcal{F}_t$ ($\mathcal{W}_t^D$). The symbol $+$ is used for frauds and $-$ for genuine transactions.

Adaptive weighting III

We can now compute some accuracy measures of $\mathcal{F}_t$ in the feedbacks requested by $\mathcal{A}_t^W$:

- $acc_{\mathcal{F}_t} = 1 - \frac{|\mathrm{ERR}_{\mathcal{F}_t}|}{k}$
- $accMiss_{\mathcal{F}_t} = 1 - \frac{|\mathrm{ERR}_{\mathcal{F}_t}| + |\mathrm{MISS}_{\mathcal{F}_t}|}{k}$
- $precision_{\mathcal{F}_t} = \frac{|\mathrm{CORR}_{\mathcal{F}_t}|}{k}$
- $recall_{\mathcal{F}_t} = \frac{|\mathrm{CORR}_{\mathcal{F}_t}|}{|\mathrm{F}_{\mathcal{A}_t^W} \cap \mathrm{Y}_t^+|}$
- $auc_{\mathcal{F}_t}$ = probability that fraudulent feedbacks rank higher than genuine feedbacks in $\mathrm{F}_{\mathcal{A}_t^W}$ according to $\mathcal{P}_{\mathcal{F}_t}$.
- $probDif_{\mathcal{F}_t}$ = difference between the mean of $\mathcal{P}_{\mathcal{F}_t}$ calculated on the frauds and the mean on the genuine in $\mathrm{F}_{\mathcal{A}_t^W}$.

If we choose $auc_{\mathcal{F}_t}$ as metric for $\mathcal{F}_t$ and similarly $auc_{\mathcal{W}_t^D}$ for $\mathcal{W}_t^D$, then α_{t+1} is:

$$\alpha_{t+1} = \frac{auc_{\mathcal{F}_t}}{auc_{\mathcal{F}_t} + auc_{\mathcal{W}_t^D}} \tag{19}$$

Selection bias and AFI

- At day t AFI provides F_t, which is not representative of B_t.
- $\forall (x, y) \in \mathrm{B}_t$ let $\mathbf{s} \in \{0, 1\}$ where $\mathbf{s} = 1$ if $(x, y) \in \mathrm{F}_t$, and $\mathbf{s} = 0$ otherwise ($\mathrm{F}_t \subset \mathrm{B}_t$).
- With $\mathcal{F}_t$ we get $\mathcal{P}_{\mathcal{K}_t}(+|x, \mathbf{s} = 1)$ rather than $\mathcal{P}_{\mathcal{K}_t}(+|x)$.
- A standard solution to SSB consist into training a weight-sensitive algorithm, where (x, y) has weight w:

$$w = \frac{\mathcal{P}(\mathbf{s} = 1)}{\mathcal{P}(\mathbf{s} = 1|x, y)} \tag{20}$$

- F_t are selected according to $\mathcal{P}(+|x)$, hence $\mathcal{P}(\mathbf{s} = 1|x)$ and $\mathcal{P}(+|x)$ are highly correlated.
- We expect $\mathcal{P}(\mathbf{s} = 1|x, y)$ to be larger for feedbacks, leading to smaller weights in (20).

INCREASING δ AND SSB CORRECTION

Table : Average P_k when $\delta = 15$.

(a) **sliding**

dataset	classifier	mean	sd
2013	$\mathcal{F}$	0.67	0.24
2013	$\mathcal{W}^D$	0.54	0.22
2013	$\mathcal{A}^W$	0.72	0.20
2014	$\mathcal{F}$	0.67	0.27
2014	$\mathcal{W}^D$	0.56	0.27
2014	$\mathcal{A}^W$	0.71	0.25
2015	$\mathcal{F}$	0.71	0.21
2015	$\mathcal{W}^D$	0.62	0.21
2015	$\mathcal{A}^W$	0.76	0.19

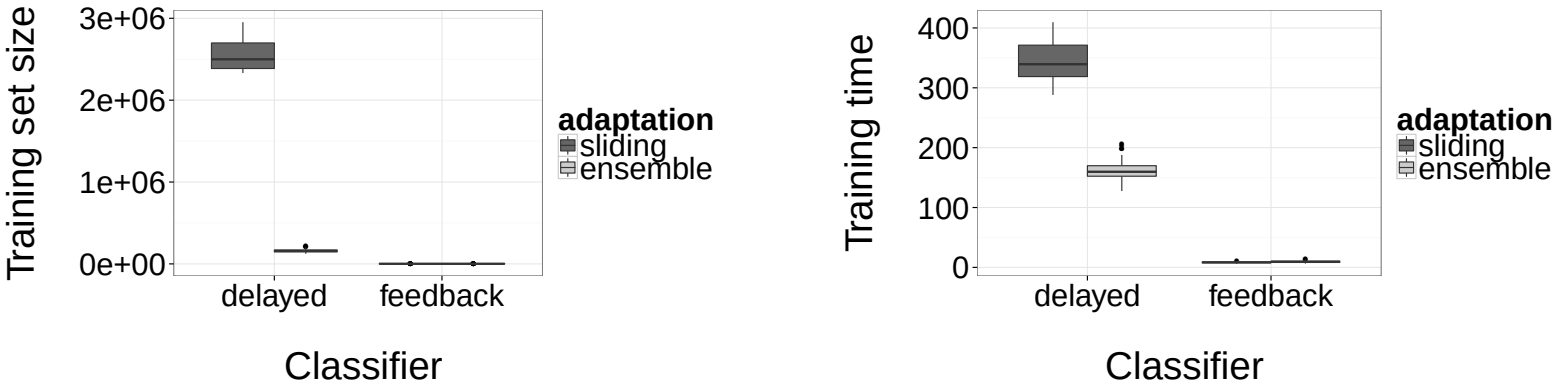
(b) **ensemble**

dataset	classifier	mean	sd
2013	$\mathcal{F}$	0.67	0.23
2013	$\mathcal{E}^D$	0.48	0.23
2013	$\mathcal{A}^E$	0.72	0.21
2014	$\mathcal{F}$	0.67	0.28
2014	$\mathcal{E}^D$	0.49	0.27
2014	$\mathcal{A}^E$	0.70	0.25
2015	$\mathcal{F}$	0.71	0.22
2015	$\mathcal{E}^D$	0.56	0.18
2015	$\mathcal{A}^E$	0.75	0.20

Table : Average P_k of $\mathcal{F}_t$ with methods for SSB correction ($\delta = 15$).

dataset	SSB correction	mean	sd
2013	SJA	0.66	0.24
2013	weigthing	0.64	0.25
2014	SJA	0.66	0.27
2014	weigthing	0.65	0.28
2015	SJA	0.71	0.22
2015	weigthing	0.68	0.23

Training set size and time



(a) Training set size

(b) Training time

Figure : Training set size and time (in seconds) to train a RF for the feedback ($\mathcal{F}_t$) and delayed classifiers ($\mathcal{W}_t^D$ and $\mathcal{E}_t^D$) in the 2013 dataset.

Feature importance

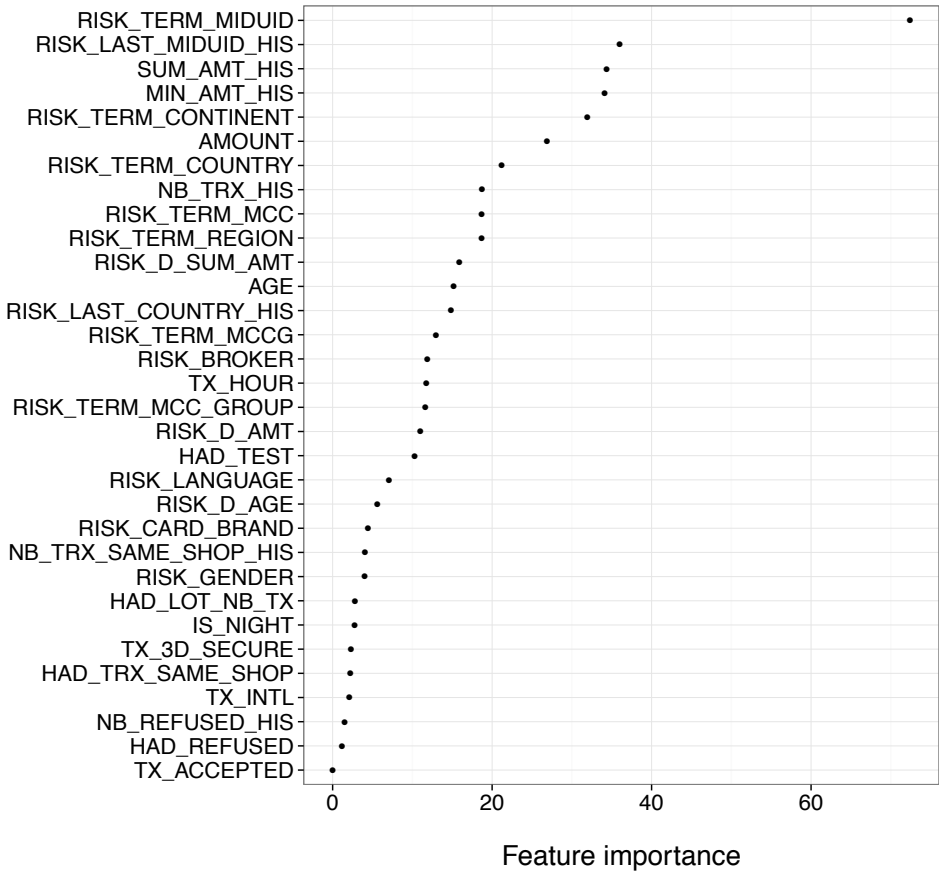


Figure : Average feature importance measured by the mean decrease in accuracy calculated with the Gini index in the RF models of $\mathcal{W}_t^D$ in the 2013 dataset. The `randomForest` package [15] available in R use the Gini index as splitting criteria.