Mitigating the Adverse Effects of Concept Drift via the Application of Reject Option

Jan Philip Göpfert[1,2], Barbara Hammer[1], and Heiko Wersing[2]

1 - Bielefeld University, Research Institute for Cognition and Robotics,
Universitätsstr. 25, 33615 Bielefeld, Germany
2 - Honda Research Institute Europe GmbH,
Carl-Legien-Str. 30, 63065 Offenbach, Germany

Abstract. Learning in non-stationary environments is challenging, because under such conditions the common assumption of independent and identically distributed data does not hold; when concept drift is present it necessitates continuous system updates. In recent years, several powerful approaches have been proposed. However, these models typically classify any input, regardless of their confidence in the classification – a strategy, which is not optimal, particularly in safety-critical environments where alternatives to a (possibly unclear) decision exist, such as additional tests or a short delay of the decision. Formally speaking, this alternative corresponds to classification with a reject option, a strategy which seems particularly promising in the context of concept drift, i.e. the occurrence of situations where the current model is wrong due to a concept change. In this contribution, we propose to extend learning under concept drift with a reject option. Specifically, we extend two recent learning architectures for drift, the self-adjusting memory architecture (SAM-kNN) and adaptive random forests (ARF), to incorporate a reject option, resulting in highly competitive state-of-the-art technologies. We evaluate their performance in learning scenarios with different types of drift.

Keywords: reject option, learning in non-stationary environments

1 Introduction

Machine learning (ML) increasingly permeates our daily lives in the form of intelligent household devices, robot companions, autonomous driving, intelligent decision support systems, fraud prevention, etc. Although ML models are getting ever more reliable – in particular due to increasing data volumes for training – they do not achieve 100 % accuracy since they rely on statistical inference. Usually, there exist situations where ML models fail and provide invalid results. Because users of a model struggle to interpret its abilities and limitations correctly [1], such failures have a measurable impact on the user's trust [2] – hence, failures should be avoided not only in safety critical environments where failures could be fatal, but also in everyday applications in order to improve user acceptance. In the case of agent models (e.g. robots), failures can often be observed easily from the agent's state (e.g. a robot not reaching its prescribed goal), and the challenge is how to communicate the cause of failure [3]. In stark contrast, failures can remain unobserved for classification models since most classifiers do not provide an explicit notion of their regime of validity. Hence the challenge arises how to enhance classifiers with an explicit notion when to reject a classification.

The notion of classification with reject option explicitly takes into account the possibility to reject a classification in unclear cases. Pioneered by Chow [4], who derived optimal reject rules if true class probabilities are known, quite a few extensions of learning with reject options have been proposed for batch learning scenarios, such as plugin rules for class probabilities [5], efficient surrogate losses [6, 7], or optimal combination schemes of local reject options [8]. These approaches deal with the classical setting of batch training based on i.i.d. data. A minor extension is offered by so-called *conformal prediction*, a framework which allows to assign probabilities to classification decisions for single inputs, and, consequently, to reject classification based on those values [9]. Here, the weaker condition of exchangeability is posed, opening the floor to online learning scenarios, but not yet to concept drift [10].

A number of approaches have been proposed for learning in non-stationary environments in the presence of concept drift, whereby several recent technologies are also suited for heterogeneous types of drift [10, 11, 12, 13, 14]. Generally speaking, concept drift is present whenever the underlying input distribution or class posterior changes, which is the case when sensors are subject to fatigue, novel and previously unseen data is observed over time, class concepts such as opinions develop over time, settings are subject to seasonal changes, etc. When learning with drift, it is almost inevitable to encounter regimes of uncertain classification – otherwise, it would not be necessary to further adapt the classification prescription, contradicting the idea of drift. Nevertheless, most learning models for non-stationary environments do not incorporate reject options. The only notable exception is the Droplets algorithm [15], which assigns some inputs explicitly to the class "reject"; size and shape of this class depend on (fixed) model meta-parameters for training. A scalable reject threshold based on the required level of certainty or user acceptance is not induced by this model.

In this contribution, we aim for an enhancement of models for learning with drift by a reject option which is based on a classifier-specific certainty measure of the classification. To the best of our knowledge, this contribution constitutes the first attempt to extend learning with drift to include reject options in such a way. The overall design implies that a suitable reject threshold can be chosen in applications. We investigate reject options for an online perceptron learning algorithm, demonstrating the complexity of the task. Afterwards, we propose a reject option for two techniques, the self-adjusting memory model and adaptive random forests, achieving convincing results. We demonstrate the benefit of learning with reject option in a couple of benchmarks which incorporate different types of drift.

2 Learning with a Reject Option

A given classifier provides a prescription $f\colon \mathbb{R}^n \to \{1, \dots, N\}$ of real-valued data to N classes. Classification with reject option extends such functionality by a special output class ϱ, which indicates that the classifier abstains from making a decision. This option is beneficial whenever the probability of a misclassification is higher than the costs for a reject. In practice, many classifiers are equipped with a certainty measure $c\colon \mathbb{R}^n \to \mathbb{R}$ which indicates the certainty of the classification, e.g. the (signed) distance to the decision boundary. In such cases, a reject strategy is often based on a simple threshold θ, i.e. the classification is of the form

$$f_\theta(x) = \begin{cases} f(x) & \text{if } c(x) \geq \theta, \\ \varrho & \text{otherwise.} \end{cases} \tag{1}$$

Provided $c(x)$ is the class probability of the output class $f(x)$, this strategy is optimal [4]. For many popular classification prescriptions, certainty measures c exist which empirically lead to excellent results [8].

2.1 Classifiers

In addition to a linear model as an initial baseline, we address an ensemble of k-NN classifiers and random forests, respectively – more complex machine learning technologies that yield state-of-the-art results. For these algorithms, the following certainty measures have been proposed:

Linear Classifier: One of the first models which has been enhanced by a reject option is the classical linear classifier. For two classes (0 and 1), a linear classifier provides the classification prescription $f(x) = H(w^\top x - \theta)$ with the Heaviside function H, an adjustable weight vector $w \in \mathbb{R}^n$ and bias θ. A typical confidence measure is offered by $c(x) = \mathrm{sgd}(w^\top x - \theta)$ with the sigmoidal $\mathrm{sgd}(t) = 1/(1 + \exp -t)$ for class 1 and $1 - \mathrm{sgd}(w^\top x - \theta)$ for class 0. This measure correlates to the distance of the data point x to the decision boundary. It has been demonstrated by Platt [16] that this form usually yields reasonable confidence measures, where – typically – slope and offset of the sigmoidal function are optimized based on the given data to enable an optimum match of its range to true confidence values.

k-NN classifier: Assume a point x is given with its k nearest neighbors $x_1, \ldots, x_k$ and corresponding labels $y_1, \ldots, y_k$. For the simple k-NN we could rely on the fraction of points of the same label within the k nearest neighbors [17]. However, this measure has the drawback that it provides $k+1$ discrete values only. A continuous extension can be based on formal grounds such as Dempster-Shafer theory [18], but this would require the tuning of several meta-parameters, rendering this measure unsuitable for online learning. Here, we rely on weighted k-NN classification instead:

$$f(x) = \mathrm{argmax}_j \left\{ \sum_{i=1}^{k} \frac{\mathbb{I}(y_i, j)}{d(x, x_i)} \;\middle|\; j = 1, \ldots, N \right\} \tag{2}$$

where

$$\mathbb{I}(y_i, j) = \begin{cases} 1, & y_i = j, \\ 0, & y_i \neq j. \end{cases} \tag{3}$$

Delany et al. [19] investigate several certainty measures and propose an accumulation of several criteria that take into account distances to closest neighbors of the same class and different classes, respectively. We approximate this value by an efficient surrogate function which can be directly derived from the weighted k-NN classification rule, the normalized average distance with values in $[0, 1]$:

$$c(x) = \left(\sum_{j=1}^{N} \sum_{i=1}^{k} \frac{\mathbb{I}(y_i, j)}{d(x, x_i)} \right)^{-1} \cdot \sum_{i=1}^{k} \frac{\mathbb{I}(y_i, \hat{y})}{d(x, x_i)}. \tag{4}$$

Random forests: Random forests as introduced by Breiman [20] constitute one of the current state-of-the-art classifiers [21], offering a classification as an ensemble of decision trees. Typically, decision trees are grown iteratively from the training data (or bootstrap samples thereof in the case of random forests), and every leaf is assigned a class probability distribution in terms of the relative frequency of the labels of the training samples assigned to this leaf. This probability can directly be interpreted as a certainty measure, but it is subject to large variance for single trees. This is greatly diminished when averaging over a bootstrap sample, as present in random forests. It has been investigated experimentally by Niculescu-Mizil and Caruana [22] that the resulting values strongly correlate to the true underlying class probabilities, hence we will use this certainty measure in the case of random forests. Its values lie within the range $[0, 1]$.

2.2 Evaluation measure

Based on the underlying class probabilities, one could obtain optimal reject strategies, but they are not known in practical applications. Good certainty measures typically strongly correlate with said probabilities, although their precise values differ [22]. An optimal choice of the threshold is often problem-dependent, reflecting the desired balance of the number of rejected data points versus the accuracy for the remaining data. As such, it is common practice to compare the efficiency of classification with reject option by a comparison of the so-called *accuracy-reject curve*: Sampling certainty thresholds $\theta \in [0, 1]$, we report the accuracy of the classification prescription for all points that are not rejected (i.e. accepted) using this threshold, together with the ratio of points that are accepted [23].

3 Learning with concept drift and its extension to reject options

In online learning, a potentially infinite stream $(\dots, (x_t, y_t), (x_{t+1}, y_{t+1}), \dots)$ of training data is given, where t denotes the current time, and each sample (x_t, y_t) is generated from an unknown probability distribution p_t. The presence of drift refers to the fact that $p_t(x, y)$ changes over time, i.e. at least two time points t_1 and t_2 exist such that $p_{t_1}(x, y) \neq p_{t_2}(x, y)$. If the posterior class probabilities change, $p_{t_1}(y|x) \neq p_{t_2}(y|x)$, we refer to real concept drift; if only the input distribution changes, $p_{t_1}(x) \neq p_{t_2}(x)$, this is referred to as virtual concept drift or covariate shift. In particular for real concept drift, a static classifier is often suboptimal, and the goal is to evolve a classification prescription h_t over time, which adjusts to the current class posterior distribution, whereby h_{t+1} is inferred from h_t and the current sample (x_t, y_t) only. The objective is to minimize the average misclassification over time as measured, for example, by the so-called *interleaved test-train error* for a time period T

$$E = \sum_{t=1}^{T} \frac{\mathbb{I}(f_t(x_t), y_t)}{T}. \tag{5}$$

This setting can be extended to online learning with reject option as soon as the classification prescription f_t is accompanied by a certainty measure c_t. In this case, given a threshold θ, classification at time point t is rejected if, and only if, $c_t(x) < \theta$. Evaluation takes place by reporting the modified interleaved test-train error

$$E_\theta = \sum_{t \leq T:\ c_t(x_t) \geq \theta} \frac{\mathbb{I}(f_t(x_t), y_t)}{|\{t \leq T\ :\ c_t(x_t) \geq \theta\}|} \tag{6}$$

and the ratio of classified data points

$$\frac{|t \leq T : c_t(x_t) \geq \theta|}{T}. \tag{7}$$

A number of learning models have been proposed which are capable of dealing with drift [10, 11, 12, 13, 14]. We address two recent models (SAM and ARF) which are suited for heterogeneous drift and which can be naturally extended to a reject option. For comparison, we look at a linear classifier (perceptron) that can adapt to drift but where useful reject strategies are problematic, as well as two sliding windows to serve as a baseline.

Online perceptron: One simple – yet popular – method, which is also available in stream mining suites such as the massive online analysis toolbox for data streams, is online perceptron learning [24]. Essentially, this consists of an online gradient descent of the squared error of the perceptron activation function $\mathrm{sgd}(w^\top x - \theta)$ based on given data with fixed step size. This model is naturally restricted to linear prescriptions, yet it yields surprisingly accurate behavior in an initial demonstration scenario as we will show in an experiment, a behavior which has also been substantiated analytically [25]. Yet, for online settings, it is not possible to adjust the sigmoidal rescaling of the perceptron output as proposed by [16], hence we will directly rely on the measure $c(x)$ for rejection as introduced above.

SAM-kNN: The Self-Adjusting Memory (SAM) architecture [26] keeps two complementary memories – short-term and long-term. The former contains the most recent samples of a data stream, whereby the length of this window is adjusted based on the classification performance, while the latter stores and continuously refines a compacted representation of previous samples as long as these are consistent with the short term memory. Depending on how the data stream changes, SAM makes flexible use of its two memories and a weighted k-nearest neighbors classifier to accurately classify even when drift is present. We extend the output of the classifier by the certainty measure as introduced above as the basis for a reject option.

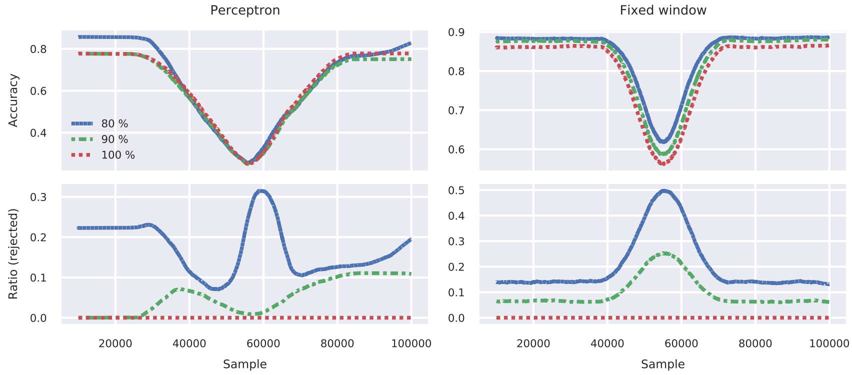

Fig. 1. Accuracy and reject ratio over time according to different reject thresholds for the perceptron and the fixed window. Thresholds are chosen such that 100 %, 90 % and 80 % of points are accepted. Accuracy and ratio are calculated over a sliding window that contains 10 % of the dataset's total number of samples.

ARF: Adaptive random forests (ARF) [14] constitute a state-of-the-art ensemble method for learning with drift. Random forests grow very fast decision trees (Hoeffding trees) online based on Poisson sampling to mimic bootstrapping effects. ARF wraps this technology into an active drift detection loop, which assigns suitable weights to an ensemble of trees, replaces unsuitable trees if drift is observed, and grows trees in the background that can serve as an intelligent initialization of such replacements when drift is expected. We can use the certainty measure as introduced for random forests above and extend it to weighted averages over the ensemble of trees as a basis for a reject option.

Sliding window: Techniques which use a classifier based on a sliding window of the data stream can serve as a baseline. We will consider a weighted k-NN classifier with a sliding window of fixed size as well as a window whose size is adapted based on the optimum classification error such as the short term memory in SAM.

4 Experiments

4.1 Linear Setting

Initially, we investigate how a perceptron's certainty responds to concept drift and demonstrate that it is not easily augmented with a reject option. To that end, we create a 2-dimensional dataset with two classes. Points are sampled uniformly from two rectangles (which determine the class label) that move towards, through, and apart from one another over time. The two classes are initially linearly separable, then become indistinguishable, and eventually become linearly separable again – albeit with a flipped separating hyperplane. Too add noise, we flip the class label of every 7th sample.

For comparison, the data is used to evaluate a fixed window[1] as well as the perceptron. The results are presented in Figure 1, with different certainty thresholds that correspond to 100 %, 90 % and 80 % of accepted (classified) points. It is apparent that the simple online perceptron is surprisingly accurate for this data set, despite its rather simple learning rule. As expected, an increasing classification difficulty is reflected in a decrease in accuracy. When the rectangles move apart (and the classes become linearly separable again), both algorithms recover. However, it is apparent that the perceptron hardly benefits from a reject option, whereas the fixed window clearly does, rejecting more points the more difficult the problem is and in such a way that the accuracy increases. Hence, it is a nontrivial task to identify effective reject options for learning with drift.

4.2 General Setting

We evaluate the efficiency of classification with reject option on a number of benchmark datasets with nonlinear characteristics. Here, model meta-paramaters are chosen in the same way as reported in Losing et al. [26] and Gomes et al. [14]. We determine accuracy reject curves by dividing the range of observed certainties into equally sized intervals and deriving the respective pareto-optimal accuracy-reject pairs. For reporting, we focus on the practically interesting range of 100 % to 50 %. We consider the benchmark datasets as described in Losing et al. [26, 12], since they cover a wide variety of different data and drift characteristics. See Table 1 for an overview over the datasets.

Table 1. Datasets considered for our experiments. Real-world datasets are followed by artificial datasets – other than that, they are presented in no particular order. Drift properties are given according to Losing et al. [12].

	# Samples	# Features	# Classes	Drift
Outdoor Objects	4000	21	40	virtual
Rialto Bridge	82 250	27	10	virtual
Poker Hand	829 201	10	10	virtual
Electricity	45 312	6	2	real
Weather	18 159	8	2	virtual
Transient Chessboard	200 000	2	8	virtual
Rotating Hyperplane	200 000	10	2	real
Interchanging RBF	200 000	2	15	real
Mixed Drift	600 000	2	15	real
Moving RBF	200 000	10	5	real
Moving Squares	200 000	2	4	real
SEA Concepts	50 000	3	2	real

[1] The fixed window serves as a straight-forward example. Results for the adaptive window, SAM, and ARF are comparable – the largest difference in accuracy between all four is below 2 %.

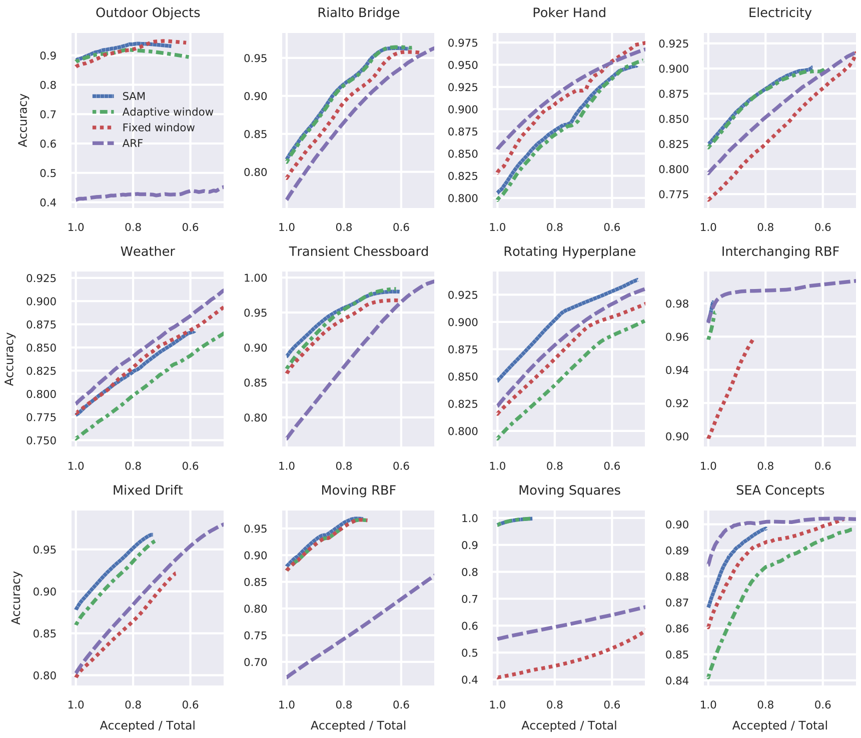

Fig. 2. Accuracy-reject curves for all datasets considered. Note the different vertical axes. The nearest neighbor based classifiers classify many samples with maximal certainty, which explains why the respective curves often end early.

Effectiveness of Reject The resulting accuracy-recject curves with respect to different certainty thresholds for all twelve datasets and all four classifiers are presented in Figure 2. As reported by Losing et al. [26] and Gomes et al. [14], it is apparent that the methods SAM and ARF are robust classifiers capable of dealing with drift, with SAM performing consistently well across all datasets considered, while ARF shows excellent results in most, but not all (in particular *Outdoor Objects*, *Moving RBF*, and *Moving Squares*). Surprisingly, also the baselines yield acceptable results for certain datasets. We observe that reject options increase the classifiers' accuracy consistently for all datasets, and influence all methods similarly: Averaged over all datasets and all four classifiers, rejecting 10 % or 20 % of all samples leads to an increase in accuracy by 3.19 % or 5.64 %, respectively. The smallest increase is 1.06 % or 1.17 %, the highest increase is 5.43 % or 10.07 %.

At present, we have used certainty measures that are intuitive and fast to compute in all cases. The curves indicate one possible weakness of these measures: in particular for k-NN classifiers (including SAM), the accuracy does not reach 100 % – rather, the curves end prematurely. This is due to the fact that k-NN assigns a certainty of 100 % to a great number of points since their k-neighborhoods are uniformly labeled. More elaborate certainty measures such as a reject option

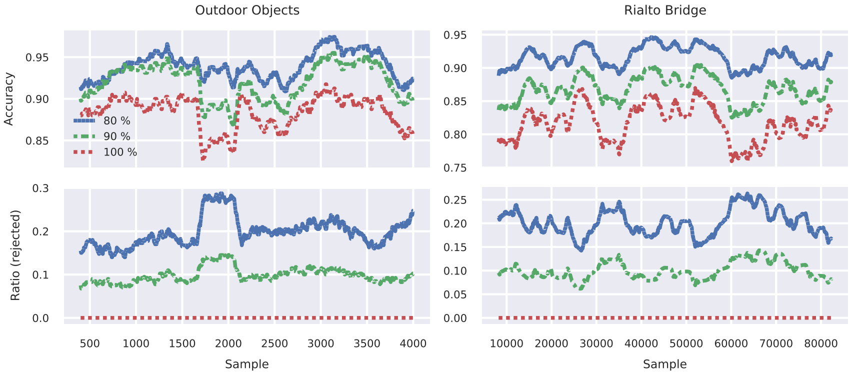

Fig. 3. Accuracy and reject ratio over time according to different reject thresholds for SAM, shown for the datasets *Outdoor Objects* and *Rialto Bridge*. Thresholds are chosen such that 100 %, 90 % and 80 % of points are accepted. Accuracy and ratio are calculated over a sliding window that contains 10 % of the respective dataset's total number of samples. Note the abrupt, temporary drop in accuracy for *Outdoor Objects* and the corresponding increase in the number of rejected points for samples 1700 to 2200.

based on absolute distances, that respects outliers, or an extension of the method to ensembles and according averaged certainties, could enable a "subtler" assessment of certainty. Hence, we see room for further improvement beyond the already satisfactory results.

Temporal Behavior As expected, accuracy varies over time in the presence of non-homogeneous drift in real-life datasets. For *Outdoor Objects* and *Rialto Bridge* we show this together with how accuracy is affected by rejection, and how the rejection ratio varies over time, when SAM with reject option is used to classify (Figure 3). Interestingly, the sharp drop in accuracy at samples 1700 to 2200 from *Outdoor Objects* is mirrored in the ratio of rejected points as a pronounced peak in rejected points. In this case, increasing the number of rejected points allows the classifier to improve so much that no notable drop in accuracy remains.

A similar – albeit less pronounced – behavior can be observed for *Rialto Bridge*. Here the overall variation in accuracy becomes much narrower. For samples 30 000 to 40 000 the abrupt loss in accuracy is compensated by more rejected points.

5 Discussion

We have introduced and evaluated diverse online learning classifiers with reject option in the presence of concept drift. Across all datasets and classifiers, we see a notable increase in accuracy when using a reject option for k-NN classifiers and ensembles of random forests. In stark contrast, reject options as presented for the perceptron do not seem easily extendable to the setting of concept drift; within an initial linear setting, reject options did not show any benefits. As expected , also for the real life non-linear data sets, no classifier achieves 100 % accuracy in the presence of drift. Interestingly, although techniques such SAM-kNN are consistently good for all settings, there is not one clear winner among the classifiers as they perform differently on various datasets. This is in line with the findings of Losing et al. [26].

Rejecting with respect to a fixed certainty threshold does not merely increase the accuracy overall but can specifically alleviate low accuracy that stems from low certainty, as seen in Figure 3. It remains to be seen how more sophisticated, time- and drift-dependent strategies for dynamically choosing certainty thresholds can improve performance even further.

Considering the particular structure of SAM, where classification depends on a choice between long- and short-term memory, it might prove beneficial to incorporate their certainties into the decision-making process – so far, it has depended solely on the memories' past performances. One must carefully investigate, however, how a classifier's certainty can be trusted, especially when the classifier performs badly in the presence of drift. On the other hand, samples with low certainty could indicate areas in which the model needs to be augmented.

As mentioned earlier, incorrect classification results can negatively impact a user's trust in a system. Because it leads to a higher accuracy, rejection alleviates these issues, but it will further be important how to communicate to a user *why* a point is rejected or – more generally – with how high a certainty a point is classified and how that certainty is to be interpreted.

References

[1] Elizabeth Cha, Anca D. Dragan, and Siddhartha S. Srinivasa. "Perceived robot capability". In: *24th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2015, Kobe, Japan, August 31 - September 4, 2015*. 2015, pp. 541–548.

[2] Munjal Desai et al. "Impact of robot failures and feedback on real-time trust". In: *HRI*. IEEE/ACM, 2013, pp. 251–258.

[3] Minae Kwon, Sandy H. Huang, and Anca D. Dragan. "Expressing Robot Incapability". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2018, Chicago, IL, USA, March 05-08, 2018*. 2018, pp. 87–95.

[4] C. Chow. "On Optimum Recognition Error and Reject Tradeoff". In: *IEEE Trans. Inf. Theor.* 16.1 (Sept. 2006), pp. 41–46. ISSN: 0018-9448.

[5] Radu Herbei and Marten H. Wegkamp. "Classification with reject option". In: *Canadian Journal of Statistics* 34.4 (2006), pp. 709–721.

[6] Peter L. Bartlett and Marten H. Wegkamp. "Classification with a Reject Option Using a Hinge Loss". In: *J. Mach. Learn. Res.* 9 (June 2008), pp. 1823–1840. ISSN: 1532-4435.

[7] Thomas Villmann et al. "Self-Adjusting Reject Options in Prototype Based Classification". In: *WSOM*. Vol. 428. Advances in Intelligent Systems and Computing. Springer, 2016, pp. 269–279.

[8] Lydia Fischer, Barbara Hammer, and Heiko Wersing. "Optimal local rejection for classifiers". In: *Neurocomputing* 214 (2016), pp. 445–457.

[9] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN: 0387001522.

[10] G. Ditzler et al. “Learning in Nonstationary Environments: A Survey”. In: *IEEE Computational Intelligence Magazine* 10.4 (Nov. 2015), pp. 12–25. ISSN: 1556-603X.

[11] Heitor Murilo Gomes et al. “A Survey on Ensemble Learning for Data Stream Classification”. In: *ACM Comput. Surv.* 50.2 (2017), 23:1–23:36.

[12] Viktor Losing, Barbara Hammer, and Heiko Wersing. “Tackling heterogeneous concept drift with the Self-Adjusting Memory (SAM)”. In: *KNOWLEDGE AND INFORMATION SYSTEMS* 54.1 (2018), pp. 171–201.

[13] Pierre-Xavier Loeffel et al. “Droplet Ensemble Learning on Drifting Data Streams”. In: *IDA*. Vol. 10584. Lecture Notes in Computer Science. Springer, 2017, pp. 210–222.

[14] Heitor Murilo Gomes et al. “Adaptive random forests for evolving data stream classification”. In: *Machine Learning* 106 (June 2017), pp. 1469–1495.

[15] P. X. Loeffel, C. Marsala, and M. Detyniecki. “Classification with a reject option under Concept Drift: The Droplets algorithm”. In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Oct. 2015, pp. 1–9.

[16] John C. Platt. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”. In: *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 1999, pp. 61–74.

[17] M. E. Hellman. “The Nearest Neighbor Classification Rule with a Reject Option”. In: *IEEE Transactions on Systems Science and Cybernetics* 6.3 (July 1970), pp. 179–185. ISSN: 0536-1567.

[18] Thierry Denoeux. “A k-nearest neighbor classification rule based on Dempster-Shafer theory”. In: *IEEE Trans. Systems, Man, and Cybernetics* 25.5 (1995), pp. 804–813.

[19] Sarah Jane Delany et al. “Generating Estimates of Classification Confidence for a Case-Based Spam Filter”. In: *Case-Based Reasoning Research and Development*. Ed. by Héctor Muñoz-Ávila and Francesco Ricci. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 177–190. ISBN: 978-3-540-31855-2.

[20] Leo Breiman. “Random Forests”. In: *Mach. Learn.* 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125.

[21] Manuel Fernández-Delgado et al. “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?” In: *Journal of Machine Learning Research* 15 (2014), pp. 3133–3181.

[22] Alexandru Niculescu-Mizil and Rich Caruana. “Predicting Good Probabilities with Supervised Learning”. In: *Proceedings of the 22Nd International Conference on Machine Learning*. ICML ’05. Bonn, Germany: ACM, 2005, pp. 625–632. ISBN: 1-59593-180-5.

[23] Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. “Accuracy-Rejection Curves (ARCs) for Comparing Classification Methods with a Reject Option”. In: *Proceedings of the third International Workshop on Machine Learning in Systems Biology*. Ed. by Sašo Džeroski, Pierre Guerts, and Juho Rousu. Vol. 8. Proceedings of Machine Learning Research. Ljubljana, Slovenia: PMLR, May 2009, pp. 65–81.

[24] Albert Bifet et al. “MOA: Massive Online Analysis”. In: *J. Mach. Learn. Res.* 11 (Aug. 2010), pp. 1601–1604. ISSN: 1532-4435.

[25] Timothy L. H. Watkin, Albrecht Rau, and Michael Biehl. “The statistical mechanics of learning a rule”. In: *Rev. Mod. Phys.* 65 (2 Apr. 1993), pp. 499–556.

[26] Viktor Losing, Barbara Hammer, and Heiko Wersing. “KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift”. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. Barcelona: IEEE, 2016, pp. 291–300.