# An empirical study about online learning with generalized passive-aggressive approaches

Adrian Perez-Suay, Francesc J. Ferri, Miguel Arevalillo-Herráez, and Jesús V. Albert

Dept. Informàtica, Universitat de València. Spain
{Adrian.Perez,Francesc.Ferri,Miguel.Arevalillo,jesus.v.albert}@uv.es

**Abstract.** This work aims at exploring different approaches to online learning that can be grouped under the common denomination of passive-aggressive techniques. In particular, we comparatively explore the original passive-aggressive formulation along with a recently proposed least-squares extension and a new proposal in which aggregate updates corresponding to small groups of labelled examples are considered instead of single samples. Preliminary results show that extended algorithms perform at least as good as basic ones in the long term but exhibit a smoother and more robust behavior along learning iterations.

## 1 Introduction

Online learning has become both an active research area and an state-of-the-art approach to cope with many challenging tasks that involve huge amounts of data. This work aims at exploring different approaches to online learning that can be grouped under the common denomination of passive-aggressive techniques [1] in the sense of the corresponding methods alternate large updates (aggressive) when predictions fail with (almost) no updates (passive) when predictions are right.

The particular online learning setting we consider here consists of a prediction algorithm that makes predictions and get updates on its underlying model at the same time. Labelled examples arrive one at a time along with corresponding labels. The predictor tries to guess the correct label and updates the model taking into account the quality of the prediction. Online learning algorithms are usually much simpler than their corresponding batch counterparts and it is often possible to derive tight bounds on their performance [2,1].

Passive-aggresive algorithms have been applied in a number of different situations [1,3]. Particularly interesting is the application this learning methodology to metric learning problems [4,5].

In this paper we comparatively explore the original passive-aggressive formulation along with a recently proposed extension that uses a Least-Squares approach instead of hinge loss minimization [5]. Moreover, a new extension of

[0] This work has been partially funded by FEDER and Spanish MEC through projects TIN2014-59641-C2-1-P, TIN2014-54728-REDC and ISIC/2012/004.

these algorithms is proposed and empirically assessed. This extension consists of performing model updates for small groups of labelled samples instead of using single examples. The motivation is to improve the robustness of the learning process when noisy examples are present. The paper is organized as follows. In the next section, the theoretical setting for passive-aggressive learning algorithms is established and the basic algorithms are summarized. Section 3 presents briefly a previous extension of the basic algorithms and introduces the new proposal along with comments about how to extend all algorithms to the nonlinear case using kernels. The empirical validation carried out in the present work is given in Section 4 and concluding remarks and further work are outlined in Section 5.

## 2 Passive-Aggressive Online Learning

Let us consider here a learning process in which (labelled) data examples come into some form of stream in such a way that only one example is given at a particular time, $t$. That is $(\ldots,(\mathbf{x}_t,y_t),\ldots)$, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1,+1\}$ in the two-class case.

In particular, the learner, that has a current model, $w_t$, receives an incoming example, $\mathbf{x}_t$ at each time step, $t$. Then, it makes a prediction, $f_t = f(w_t;\mathbf{x}_t)$, which is converted into a predicted label using the sign function, $\hat{y_t} = \mathrm{sgn}(f_t)$. At this moment, the learner compares both labels and prediction using an appropriate loss function which is in turn used to decide whether or not and how to update the current model, $w_t$.

We will restrict ourselves to linear prediction models in which the model is given by a vector weight and an explicit bias, $(\mathbf{w},b)$. Then the prediction is computed as $f((\mathbf{w},b);\mathbf{x}) = \mathbf{w}^\top\mathbf{x} - b$.

Passive-aggressive methods are obtained when each update leads to the closest model that that minimizes the so-called hinge loss. In the general non-separable case, this leads to a minimization problems as [1]

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|_2^2 + \frac{1}{2}(b-b_t)^2 + C\xi \tag{1}$$

$$s.t. \quad \begin{cases} y_t(\mathbf{w}^\top\mathbf{x}_t - b) \geq 1-\xi \\ \xi \geq 0 \end{cases} \tag{2}$$

Alternatively, the penalty term, $C\xi$, can be squared and then the positivity constraint in 2 can be safely dropped. These two minimization problems can be tackled using Lagrange multipliers to arrive to a closed solution that is given as the following update equations

$$\mathbf{w} = \mathbf{w}_t + \tau y_t \mathbf{x}_t, \tag{3}$$

$$b = b_t - \tau y_t, \tag{4}$$

in which the learning rate, $\tau$, is given in terms of the hinge loss, $\ell_t = \max\{0, 1-y_t f_t\}$ $= \max\{0, 1-y_t(\mathbf{w}_t^\top\mathbf{x}_t - b_t)\}$ as

$$\tau = \min\left\{C, \frac{\ell_t}{1 + \|\mathbf{x}_t\|^2}\right\} \tag{5}$$

for the minimization problem in Equations 1 and 2, and as

$$\tau = \frac{\ell_t}{1 + \|\mathbf{x}_t\|^2 + \frac{1}{2C}} \tag{6}$$

for the alternative formulation that uses a squared penalty term. The two resulting algorithms are known as Passive-aggressive I (PAI) and II, (PAII), respectively [1].

## 3 Generalized Passive-Aggressive Methods

### 3.1 Least-Squares Online Learning

Instead of trying to minimize the hinge loss at each step, least-squares can be alternatively adopted [5]. The corresponding formulation which has been referred to as PALS, is very similar to the one for PAII but using an equality constraint. In particular, the corresponding minimization problem is

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2 + \frac{1}{2}(b - b_t)^2 + C\xi^2 \tag{7}$$

$$s.t. \quad y_t(\mathbf{w}^\top \mathbf{x}_t - b) = 1 - \xi \tag{8}$$

This problem can be tackled exactly as the basic passive-aggressive formulation. In fact, the same update Equations 3 and 4 are obtained and the corresponding expression for the learning rate is

$$\tau = \frac{1 - f_t}{1 + \|\mathbf{x}_t\|^2 + \frac{1}{2C}} \tag{9}$$

Note that the main difference is that the term in the numerator can be either positive (when prediction fails) or negative (when prediction is correct) thus leading to aggressive updates regardless on the correctness of the prediction. Passive steps are consequently reduced to prediction values that fall on the (soft) margin of the corresponding linear predictor.

### 3.2 Passive-Aggressive using Mini-batches

Although passive-aggressive learning exhibits in general a very good behavior in many different scenarios, it may sometimes lead to erratic model updates when noisy examples are present. In order to reduce the isolated effect of updates due to outliers it is relatively natural to think of aggregating the effect of several updates into one single step. The idea of updating learning models using mini-batches has been already in other previous works [6] With this rationale in mind,

we propose here to reformulate passive-aggressive online learning for small groups of labelled examples. The resulting generalized learning schemes will be referred to here as BPAI, BPAII and BPALS, respectively.

At each time, $t$, we have a set of $B$ (labelled) examples, $\{(\mathbf{x}_{t-k}, y_{t-k})\}_{k=0}^{B-1}$. If we extend the previous passive-aggressive approaches to this new setting, we can arrive at the following minimization problem:

$$\min_{\mathbf{w},b,\xi_k} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|_2^2 + \frac{1}{2}(b-b_t)^2 + C\sum_k \xi_k \tag{10}$$

$$s.t. \quad \left\{ \begin{array}{c} \overline{y}_k(\mathbf{w}^\top \overline{\mathbf{x}}_k - b) \geq 1-\xi_k \\ \xi_k \geq 0 \end{array} \right\} \; k=0,1,\ldots,B-1 \tag{11}$$

where we have defined $\overline{\mathbf{x}}_k = \mathbf{x}_{t-k}$ and $\overline{y}_k = y_{t-k}$.

Introducing Lagrange multipliers and equating the corresponding derivatives of the Lagrangian to zero as with the previous (one-example) problems [1] we can arrive at the new updating equations

$$\mathbf{w} = \mathbf{w}_t + \sum_k \tau_k \overline{y}_k \overline{\mathbf{x}}_k, \tag{12}$$

$$b = b_t - \sum_k \tau_k \overline{y}_k, \tag{13}$$

along with the condition $\tau_k \leq C$ for $k=0,\ldots,B-1$.

By substituting back into the Lagrangian we arrive at the following dual maximization problem in terms of $\boldsymbol{\tau} = (\tau_0,\ldots,\tau_{B-1})^\top$

$$\max_{\boldsymbol{\tau}} \quad -\frac{1}{2}\sum_{j,k}\tau_j\tau_k\overline{y}_j\overline{y}_k(\overline{\mathbf{x}}_j{}^\top\overline{\mathbf{x}}_k+1) + \sum_k \tau_k(1-\overline{y}_k\overline{f}_k) \tag{14}$$

$$s.t. \quad 0 \leq \tau_k \leq C, \quad k=0,\ldots,B-1 \tag{15}$$

where $\overline{f}_k = \mathbf{w}_t{}^\top\overline{\mathbf{x}}_k - b_t = \mathbf{w}_t{}^\top\mathbf{x}_{t-k} - b_t$.

The criterion in Equation 14 can be written also in matrix form as

$$\max_{\boldsymbol{\tau}} \quad -\frac{1}{2}\boldsymbol{\tau}^\top A\boldsymbol{\tau} + \boldsymbol{\tau}^\top\boldsymbol{\ell} \tag{16}$$

where $\boldsymbol{\ell} = (\overline{\ell}_0,\ldots,\overline{\ell}_{B-1})$, $\overline{\ell}_k = 1-\overline{y}_k\overline{f}_k$, and $A_{jk} = \overline{y}_j\overline{y}_k(\overline{\mathbf{x}}_j{}^\top\overline{\mathbf{x}}_k+1)$.

If we proceed in the same way for the passive-aggressive variant that uses quadratic penalty, we arrive to a slightly different dual problem

$$\max_{\boldsymbol{\tau}} \quad -\frac{1}{2}\boldsymbol{\tau}^\top A\boldsymbol{\tau} - \frac{1}{2C}\boldsymbol{\tau}^\top\boldsymbol{\tau} + \boldsymbol{\tau}^\top\boldsymbol{\ell} \tag{17}$$

$$s.t. \quad \tau_k \geq 0, \quad k=0,\ldots,B-1 \tag{18}$$

Both dual problems are standard quadratic optimization problems that can be appropriately tackled with any off-the-shelf method taking into account that $B$ is relatively small in practice.

If we repeat the previous derivation for the corresponding Least-squares formulation, we arrive at the same criterion in Equation 17, but in this case we get an unconstrained optimization problem. Consequently, by differentiation with respect to $\boldsymbol{\tau}$ and equating to zero we arrive at the following closed expression

$$\boldsymbol{\tau} = (A + \tfrac{1}{C} I)^{+} \boldsymbol{\ell} \tag{19}$$

where $I$ is the identity matrix and the superscript $^{+}$ refers to the Moore-Penrose pseudoinverse. Note that by definition, the resulting matrix is symmetric and semidefinite positive and correspondingly, easily (pseudo) invertible.

### 3.3 Kernel extensions

All the previous algorithms including the new proposals, can be entirely formulated in terms of inner products between input vectors that can be safely replaced with any general Mercer kernel [1]. The corresponding kernelized algorithms use only the coefficients in $\boldsymbol{\tau}$ at each step.

The main problem when implementing these kernelized versions is that the number of support vectors grows linearly with the number of examples taken into account. Although methods for appropriately moderating this growth have been proposed [7], they have not been taken into account in the present paper.

## 4 Experiments

In order to compare the goodness and main properties of all the algorithms considered including the new proposal, a set of 7 two-class datasets have been selected from very well-known and widely used public databases [8,9]. In particular, the selected databases are shown in Table 1 along with their particular number of elements and dimensionality.

**Table 1.** Databases used in the experiments along with total number of elements and dimensionality.

| name | ♯ | size | dimensionality |
|---|---|---|---|
| breast | 1 | 683 | 9 |
| diabetes | 2 | 768 | 8 |
| heart | 3 | 297 | 13 |
| ionosphere | 4 | 351 | 34 |
| liver | 5 | 345 | 6 |
| sonar | 6 | 208 | 60 |
| twonorm | 7 | 7400 | 20 |

The size of mini-batches, i.e. the parameter $B$, is the most critical parameter. For the exploratory experimentation carried out in the present study, values of

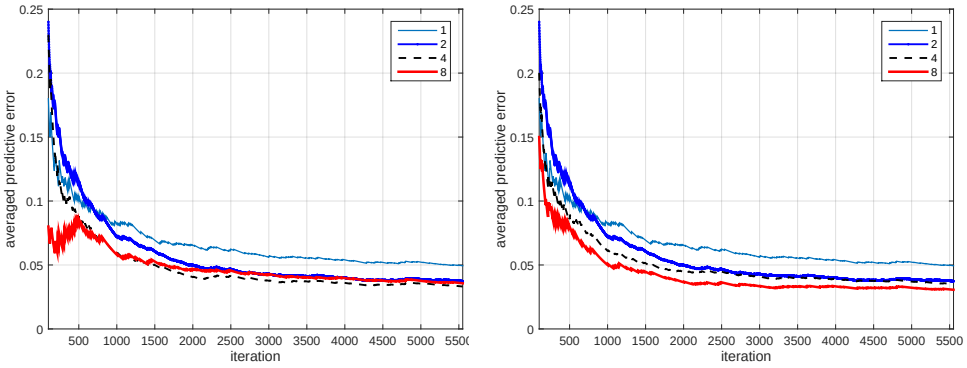


**Fig. 1.** Averaged cumulative prediction error corresponding to BPAI and BPALS for one learning run on the database twonorm. Results for different values of $B$ are shown.

$B$ below 10 have been considered. Moreover, the isolated value of $B = 20$ is also included in the experiments to check whether results keep improving or not.

To properly assess the capabilities of the online algorithms, we consider the online predictive averaged error as in most similar studies about online learning. This measure gives an estimation along the online learning process of how many times the model makes a wrong prediction about the arriving sample or group of samples. Moreover, classification results using a different test set are also measured to illustrate how online learned models behave on unseen data.

### 4.1 Experimental Setup

The particular experimental setup is very similar to the protocols used in other studies about online learning. Firstly, we consider the task of estimating appropriate values for the aggressiveness parameter, $C$, and the RBF kernel parameter, $\sigma$. In the present work, these values have been selected taking the best averaged results of the final averaged online predictive error over a fixed set of 7 logarithmically spaced values for $C \in [10^{-5}, 10]$ and 6 logarithmically spaced values of $\sigma \in [10^{-4}, 10]$ using 3 random subsamples of the available data. Then, the values corresponding to the lowest averaged error rate have been fixed for the training task.

In the training task, 25 independent learning trials have been run using 75% of the available data in each database. Finally, the test task consists in measuring the final performance of the models learned in the training step on previously unseen data, i.e. the remaining 25%.

### 4.2 Empirical results and discussion

In order to illustrate the training step, the averaged predictive errors for each learning iteration for two algorithms in a particular learning trial on the database

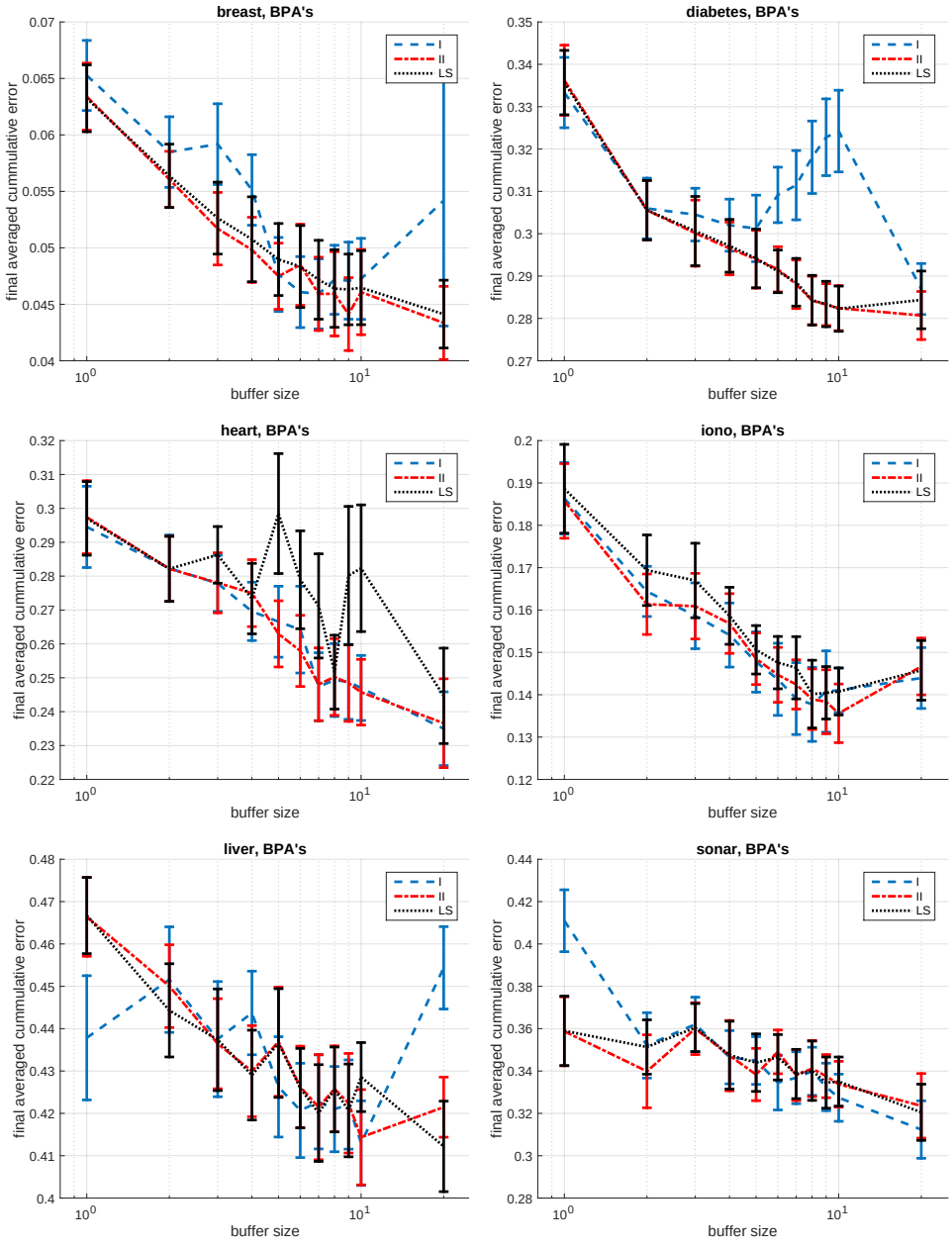


**Fig. 2.** Averaged overall cumulative prediction error for 6 of the databases considered. mini-batch size, $B$, is shown in logarithmic scale.

twonorm are shown in Figure 1 for some of the considered values of $B$. These curves look very similar for all databases and all three variants (BPAI, BPAII, and BPALS) with regard to the dependency on the parameter $B$.

As a global measure of performance, one can take the final averaged predictive error. That is, the prediction error corresponding to the last learning iteration. Figure 2 shows this final predictive error for 6 of the seven databases considered in the present work. The curves corresponding to the twonorm database are basically identical to the ones for diabetes (with different absolute values) and are not shown.

Averaged classification error rates for some values of $B$ are also measured and are shown in Table 2 for all the databases considered. These error rates are computed using the last trained predictor (at the end of the learning process). The final model is the one which has used the maximum amount of data and is expected to generalize well on unseen data.

The best results in terms of classification rates are obtained for values of $B = 4, 8$. The methods which offer the best results are BPAI ($B = 4$) and BPALS ($B = 8$). The methods BPAI ($B = 8$) and BPAII ($B = 4$) and BPAII ($B = 8$) lead also to very good results. Nevertheless, results with regard to classification errors are all very similar if we take into account the small differences obtained and corresponding statistical significances.

With regard to averaged cumulative errors shown in Figure 2, we can observe more significant differences among different approaches. As a common feature, the predictive error decreases as $B$ increases with an apparent linear tendency. It can also be noted that there are significant differences between each method with $B = 1$ and sufficiently large values of $B$. In several databases, even $B = 2$ leads to significant differences. With regard to different minimization schemes, usually both BPAII and BPALS exhibit a very similar behavior while BPAI is usually worse but also better for some databases. As an interesting general fact, the methods keep performing well if we increase the mini-batch size up to 20 in most of the databases and for most of the algorithms.

In general, it can be said that the minimum value of $B$ for which significantly better results are obtained lies somewhere in the interval [4,8]. In this range of values, the algorithms have a computational burden which is only slightly above the one for the basic algorithm (i.e. $B = 1$). This computational burden is significantly smaller in the case of BPALS as the corresponding optimal value can be computed in a more straightforward way.

**Table 2.** Averaged classification error for all databases considered. Confidence intervals at 95% are shown in parenthesis.

| ♯ | PAI-1 | BPAI-4 | BPAI-8 | PAII-1 | BPAII-4 | BPAII-8 | PALS-1 | BPALS-4 | BPALS-8 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 6.56(1.30) | **4.42(0.58)** | 5.62(0.60) | 6.31(1.53) | 5.65(0.68) | 4.87(0.64) | 6.59(1.67) | 4.56(0.61) | 4.89(0.65) |
| 2 | 25.60(1.24) | 24.56(1.15) | 25.92(1.62) | 26.12(1.68) | 23.94(1.14) | 23.96(1.19) | 26.65(1.78) | 24.04(1.19) | **23.83(1.16)** |
| 3 | 29.35(6.78) | 20.00(1.84) | 19.30(1.52) | 21.62(2.29) | 19.57(1.56) | 18.97(1.48) | 21.57(2.24) | 19.62(1.72) | **18.54(1.52)** |
| 4 | 21.70(2.13) | 23.68(1.59) | 25.06(1.39) | 21.29(1.96) | 23.03(1.55) | 23.77(1.57) | **21.06(2.51)** | 22.25(1.94) | 23.54(1.93) |
| 5 | 41.30(2.34) | 42.05(1.76) | **39.63(1.83)** | 42.84(3.30) | 40.00(1.86) | 39.95(2.01) | 43.02(3.24) | 40.19(1.61) | 40.37(2.02) |
| 6 | 16.47(2.34) | **13.10(1.34)** | 13.18(1.47) | 14.98(2.55) | 14.20(1.46) | 13.25(1.54) | 14.98(2.55) | 14.20(1.39) | 13.65(1.61) |
| 7 | 2.65(0.23) | **2.30(0.15)** | **2.30(0.14)** | 2.64(0.19) | **2.30(0.14)** | **2.30(0.14)** | 2.64(0.19) | 2.31(0.15) | **2.30(0.14)** |

## 5 Concluding Remarks

A comparative study about different strategies for online learning under the common denomination of passive-aggressive methods has been presented. Moreover, a new proposal that consists of considering joint updates corresponding to small groups of labelled samples has been introduced. According to the preliminary experimentation carried out, very promising results for some combined extensions of the basic algorithms have been obtained at moderate computational burden.

Further work is being done on applying this algorithm to metric learning problems. Fine tuning of parameters to arrive to an optimal trade-off between performance and computational cost is also under consideration.

## References

1. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research **7** (2006) 551–585
2. Kivinen, J., Warmuth, M.K.: Exponentiated gradient versus gradient descent for linear predictors. Inf. Comput. **132** (1997) 1–63
3. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. Journal of Machine Learning Research **11** (2010) 1109–1135
4. Shalev-Shwartz, S., Singer, Y., Ng, A.Y.: Online and batch learning of pseudo-metrics. In: Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004. (2004)
5. Pérez-Suay, A., Ferri, F.J., Arevalillo-Herráez, M.: Passive-aggressive online distance metric learning and extensions. Progress in AI **2** (2013) 85–96
6. Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for svm. Mathematical Programming **127** (2011) 3–30
7. Crammer, K., Kandola, J., Singer, Y.: Online classification on a budget. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16. MIT Press (2004) 225–232
8. Lichman, M.: UCI machine learning repository (2013)
9. Duin, R.P.W.: Prtools version 3.0: A matlab toolbox for pattern recognition. In: Proc. of SPIE. (2000) 1331