Online F-Measure Optimization

Authored by:

Eyke H?llermeier
Bal?zs Sz?r?nyi
R?bert Busa-Fekete
Krzysztof Dembczynski

Abstract

The F-measure is an important and commonly used performance metric for binary prediction tasks. By combining precision and recall into a single score, it avoids disadvantages of simple metrics like the error rate, especially in cases of imbalanced class distributions. The problem of optimizing the F-measure, that is, of developing learning algorithms that perform optimally in the sense of this measure, has recently been tackled by several authors. In this paper, we study the problem of F-measure maximization in the setting of online learning. We propose an efficient online algorithm and provide a formal analysis of its convergence properties. Moreover, first experimental results are presented, showing that our method performs well in practice.

1 Paper Body

Being rooted in information retrieval [16], the so-called F-measure is nowadays routinely used as a b = (b performance metric in various prediction tasks. Given predictions y y1 , . . . , ybt) 2 {0, 1}t of t binary labels y = (y1 , . . . , yt), the F-measure is defined as Pt b) ? recall(y, y b) 2 i=1 yi ybi 2 ? precision(y, y b) = Pt F (y, y = 2 [0, 1] , (1) Pt b b precision(y, y) + recall(y, y) y + y b i=1 i i=1 i Pt Pt Pt Pt b) = b) = where precision(y, y bi / i=1 ybi , recall(y, y bi / i=1 yi , and where i=1 yi y i=1 yi y 0/0 = 1 by definition. Compared to measures like the error rate in binary classification, maximizing the F-measure enforces a better balance between performance on the minority and majority class; therefore, it is more suitable in the case of imbalanced data. Optimizing for such an imbalanced measure is very important in many real-world applications where positive labels are significantly less frequent than negative ones. It can also be generalized to a weighted harmonic average of precision and recall. Yet, for the sake of simplicity, we stick to the unweighted mean, which is often referred to as the F1-score or the F1-measure.

Given the importance and usefulness of the F-measure, it is natural to look for learning algorithms that perform optimally in the sense of this measure. However, optimizing the F-measure is a quite challenging problem, especially because the measure is not decomposable over the binary predictions. This problem has received increasing attention in recent years and has been tackled by several authors [19, 20, 18, 10, 11]. However, most of this work has been done in the standard setting of batch learning. 1

In this paper, we study the problem of F-measure optimization in the setting of online learning [4, 2], which is becoming increasingly popular in machine learning. In fact, there are many applications in which training data is arriving progressively over time, and models need to be updated and maintained incrementally. In our setting, this means that in each round t the learner first outputs a prediction ybt and then observes the true label yt . Formally, the protocol in round t is as follows: 1. first an instance xt 2 X is observed by the learner, 2. then the predicted label ybt for xt is computed on the basis of the first t instances (x1 , . . . , xt), the t 1 labels (y1 , . . . , yt 1) observed so far, and the corresponding predictions (b y1 , . . . , ybt 1), 3. finally, the label yt is revealed to the learner.

The goal of the learner is then to maximize

F(t) = F ((y1 , . . . , yt), (b y1 , . . . , ybt))

(2)

over time. Optimizing the F-measure in an online fashion is challenging mainly because of the non-decomposability of the measure, and the fact that the ybt cannot be changed after round t.

As a potential application of online F-measure optimization consider the recommendation of news from RSS feeds or tweets [1]. Besides, it is worth mentioning that online methods are also relevant in the context of big data and large-scale learning, where the volume of data, despite being finite, prevents from processing each data point more than once [21, 7]. Treating the data as a stream, online algorithms can then be used as single-pass algorithms. Note, however, that single-pass algorithms are evaluated only at the end of the training process, unlike online algorithms that are supposed to learn and predict simultaneously. We propose an online algorithm for F-measure optimization, which is not only very efficient but also easy to implement. Unlike other methods, our algorithm does not require extra validation data for tuning a threshold (that separates between positive and negative predictions), and therefore allows the entire data to be used for training. We provide a formal analysis of the convergence properties of our algorithm and prove its statistical consistency under different assumptions on the learning process. Moreover, first experimental results are presented, showing that our method performs well in practice.

2

Formal Setting

In this paper, we consider a stochastic setting in which (x1 , y1), . . . , (xt , yt) are assumed to be i.i.d. samples from some unknown distribution ?(?) on X ? Y, where Y = {0, 1} is the label space and X is some instance space. We denote the marginal distribution of the feature vector X by ?(?).1 Then, the posterior probability of the positive class, i.e., the conditional probability that Y = 1 ?(x,1) given X = x, is ?(x) = P(Y = 1 — X = x) = ?(x,0)+?(x,1) . The prior distribution of class 1 can R be written as ?1 = P(Y = 1) = x2X ?(x) d?(x). Let B = {f : X ! {0, 1}} be the set of all binary classifiers over the set X . The F-measure of a binary classifier f 2 B is calculated as R 2 X ?(x)f (x) d?(x) 2E [?(X)f (X)] R F (f) = R = . E [?(X)] + E [f (X)] ?(x) d?(x) + f (x) d?(x) X X

According to [19], the expected value of (1) ?converges to F (f) with t ! 1 when ? f is used to calculate yb, i.e., ybt = f (xt). Thus, limt!1 E F (y1 , . . . , yt), (f (x1), . . . , f (xt)) = F (f).

Now, let G = {g : X ! [0, 1]} denote the set of all probabilistic binary classifiers over the set X , and let T ? B denote the set of binary classifiers that are obtained by thresholding a classifier g 2 G?that is, classifiers of the form g ? (x) = Jg(x)

?K

(3)

for some threshold ? 2 [0, 1], where J?K is the indicator function that evaluates to 1 if its argument is true and 0 otherwise. 1

X is assumed to exhibit the required measurability properties.

2

According to [19], the optimal F-score computed as maxf 2B F (f) can be achieved by a thresholded classifier. More precisely, let us define the thresholded F-measure as R 2 X ?(x) J?(x) ? K d?(x) 2E [?(X) J?(X) ? K] ? R F (?) = F (?) = R = (4) E [?(X)] + E [J?(X) ? K] ?(x) d?(x) + J?(x) ? K d?(x) X X Then the optimal threshold ? ? can be obtained as

? ? = argmax F (?) .

(5)

0?? ?1

Clearly, for the classifier in the form of (3) with g(x) = ?(x) and ? = ? ? , we have F (g ?) = F (? ?). Then, as shown by [19] (see their Theorem 4), the performance of any binary classifier f 2 B cannot exceed F (? ?), i.e., F (f) ? F (? ?) for all f 2 B. Therefore, estimating posteriors first and adjusting a threshold afterward appears to be a reasonable strategy. In practice, this seems to be the most popular way of maximizing the F-measure in a batch mode; we call it the 2-stage F-measure maximization approach, or 2S for short. More specifically, the 2S approach consists of two steps: first, a classifier is trained for estimating the posteriors, and second, a threshold is tuned on the posterior estimates. For the time being, we are not interested in the training of this classifier but focus on the second step, that is, the labelling of instances via thresholding posterior probabilities. For doing this, suppose a finite set DN = {(xi , yi)}N i=1 of labeled instances are given as training information. Moreover, suppose estimates pbi = g(xi) of the posterior probabilities pi = ?(xi) are provided by a classifier g 2 G. Next, one might define the F-score obtained by applying the threshold classifier g ? on the data DN as follows: PN yi J? ? g(xi)K F (? ; g, DN) = PN i=1 PN (6) y + i=1 i i=1 J? ? g(xi)K In order to find an optimal threshold ?N 2 argmax0?? ?1 F (? ; g, DN), it suffices to search the P

finite set {b p1 , . . . , pbN }, which requires time O(N log N). In [19], it is shown that F (? ; g, DN) ! F (g ?) as N ! 1 for any ? 2 (0, 1), and [11] provides an even stronger result: If a classifier gDN is induced from DN by an L1 -consistent learner,2 and a threshold ?N is obtained by maximizing (6) P ?N 0 on an independent set DN , then F (gD) ! F (? ?) as N ! 1 (under mild assumptions on the N data distribution).

3

Maximizing the F-Measure on a Population Level

In this section we assume that the data distribution is known. According to the analysis in the previous section, optimizing the F-measure boils down to finding the optimal threshold ? ? . At this point, an observation is in order. Remark 1. In general, the function F (?) is neither convex nor concave. For example, when X is finite, then the denominator and enumerator of (4) are step functions, whence so is F (?). Therefore, gradient methods cannot be applied for finding ? ? . Nevertheless, ? ? can be found based on a recent result of [20], who show that finding the root of Z h(?) = max (0, ?(x) ?) d?(x) ? ?1 (7) x2X

is a necessary and sufficient condition for optimality. Note that h(?) is continuous and strictly decreasing, with h(0) = ?1 and h(1) = ?1 . Therefore, h(?) = 0 has a unique solution which is ? ? . Moreover, [20] also prove an interesting relationship between the optimal threshold and the F-measure induced by that threshold: F (? ?) = 2? ? . The marginal distribution of the feature vectors, ?(?), induces a distribution ?(?) on the posteriors: R ?(p) = x2X J?(x) = pK d?(x) for all p 2 [0, 1]. By definition, J?(x) = pK is the Radon-Nikodym derivative of d? d? , and ?(p) the density of observing an instance x for which the probability of the 2

A learning algorithm, viewed as a map?from samples DN to classifiers gDN ? , is called L1 -consistent w.r.t. R the data distribution ? if limN !1 PDN ?? x2X —gDN (x) ?(x)— d?(x) ¿ ? = 0 for all ? ¿ 0.

3

positive label is p. We shall write concisely d?(p) = ?(p) dp. Since ?(?) is an induced probability measure, the measurable transformation allows us to rewrite the notions introduced above in terms of ?(?) instead of ?(?)?see, for example, Section 1.4 in [17]. For example, the prior probability R R1 ?(x) d? can be written equivalently as 0 p d?(p). Likewise, (7) can be rewritten as follows: X Z 1 Z 1 Z 1 Z 1 h(?) = max (0, p ?) d?(p) ? p d?(p) = p ? d?(p) ? p d?(p) 0 0 ? 0 ?Z 1 Z 1 Z 1 = p d?(p) ? 1 d?(p) p d?(p) (8) ?

?

0

Equation (8) will play a central role in our analysis. Note that precise knowledge of ?(?) suffices to find the maxima of F (?). This is illustrated by two examples presented in Appendix E, in which we assume specific distributions for ?(?), namely uniform and Beta distributions.

4

Algorithmic Solution

In this section, we provide an algorithmic solution to the online F-measure maximization problem. For this, we shall need in each round t some classifier gt 2 G that provides us with some estimate pbt = gt (xt) of the probability ?(xt). We would like to stress again that the focus of our analysis is on optimal thresholding instead of classifier learning. Thus, we assume the sequence of classifiers g1 , g2 , . . . to be produced by an external online learner, for example, logistic regression trained by stochastic gradient descent. As an aside, we note that F-measure maximization is not directly comparable with the task that is most often considered and analyzed in online learning, namely regret minimization [4]. This is mainly because the F-measure is a non-decomposable performance metric. In fact, the cumulative regret is a summation of a per-round regret rt , which only depends on the prediction ybt and the true outcome yt [11]. In the case of the F-measure, the score F(t) , and therefore the optimal prediction ybt , depends on the entire history, that is, all observations and decisions made by the learner till time t. This is discussed in more detail in Section 6. The most naive way of forecasting labels is to implement online learning as repeated batch learning, that is, to apply a batch learner (such as 2S) to Dt = {(xi , yi)}ti=1 in each time step t. Obviously, however, this strategy is prohibitively expensive, as it requires storage of all data points seen so far (at least in mini-batches), as well as optimization of the threshold ?t and re-computation of the classifier gt on an ever growing number of examples.

In the following, we propose a more principled technique to maximize the online F-measure. Our approach is based on the observation that h(? ?) = 0 and h(?)(? ? ?) ¡ 0 for any ? 2 [0, 1] such that ? 6= ? ? [20]. Moreover, it is a monotone decreasing continuous function. Therefore, finding the optimal threshold ? ? can be viewed as a root finding problem. In practice, however, h(?) is not known and can only be estimated. Let us define h ?, y, yb = yb y ? (y + yb) . For now, assume ?(x) b to be known and write concisely h(?) = h(?, y, J?(x) ? K). We can compute the expectation of b h(?) with respect to the data distribution for a fixed threshold ? as follows: h i E b h(?) = E [h(?, y, J?(x) ? K)] = E [y J?(x) ? K ? (y + J?(x) ? K)] ?Z 1 ? Z 1 = p Jp ? K d?(p) ? p + Jp ? K d?(p) =

Z

0

1

p d?(p)

?

?

?Z

0

1

p d?(p) +

0

Z

1

(9)
1 d?(p) = h(?)
?
Thus, an unbiased estimate of h(?) can be obtained by evaluating b h(?) for an instance x. This suggests designing a stochastic approximation algorithm that is able to find the root of h(?) similarly to the Robbins-Monro algorithm [12]. Exploiting the relationship between the optimal F-measure and the optimal threshold, F (? ?) = 2? ? , we define the threshold in time step t as t
?t =
X 1 at F(t) = where at = yi ybi , 2 bt i=1 4
bt =
t X i=1
yi +
t X i=1
ybi .
(10)
With this threshold, the first differences between thresholds, i.e. ?t+1 ?t , can be written as follows. Proposition 2. If thresholds ?t are defined according to (10) and ybt+1 as J?(xt+1) ¿ ?t K, then (?t+1
?t)bt+1 = h(?t , yt+1 , ybt+1) .
(11)
The proof of Prop. 2 is deferred to Appendix A. According to (11), the method we obtain ?almost? coincides with the update rule of the Robbins-Monro algorithm. There are, however, some notable differences. In particular, the sequence of coefficients, namely the values 1/bt+1 , does not consist of predefined real values converging to zero (as fast as 1/t). Instead, it consists of random quantities that depend on the history, namely the observed labels y1 , . . . , yt and the predicted labels yb1 , . . . , ybt . Moreover, these ?coefficients? are not independent of h(?t , yt+1 , ybt+1) either. In spite of these additional difficulties, we shall present a convergence analysis of our algorithm in the next section.

The pseudo-code of our online F-measure optimization algorithm, called Online F-measure Algorithm 1 OFO Optimizer (OFO), is shown in Algorithm 1. 1: Select g0 from B, and set ?0 = 0 The forecast rule can be written in the form of 2: for t = 1 ! 1 do Observe the instance xt ybt = Jpt ?t 1 K for xt where the threshold is 3: pbt gt 1 (xt) . estimate posterior defined in (10) and pt = ?(xt). In practice, we 4: ybt Jb pt ? t 1 K . current prediction use pbt = gt 1 (xt) as an estimate of the true 5: Observe label yt posterior pt . In line 8 of the code, an online 6: at t Calculate F(t) = 2a learner A : G ? X ? Y ! G is assumed, 7: bt and ?t = bt which produces classifiers gt by incrementally 8: gt A(gt 1 , xt , yt) . update the classifier updating the current classifier with the newly 9: return ?T observed example, i.e., gt = A(gt 1 , xt , yt). In our experimental study, we shall test and compare various state-of-the-art online learners as possible choices for A.
5
Consistency

In this section, we provide an analysis of the online F-measure optimizer proposed in the previous section. More specifically, we show the statistical consistency of the OFO algorithm: The sequence of online thresholds and F-scores produced by this algorithm converge, respectively, to the optimal threshold ? ? and the optimal thresholded F-score F (? ?) in probability. As a first step, we prove this result under the assumption of knowledge about the true posterior probabilities; then, in a second step, we consider the case of estimated posteriors. Theorem 3. Assume the posterior probabilities pt = ?(xt) of the positive class to be known in each step of the online learning process. Then, the sequences of thresholds ?t and online F-scores F(t) produced by OFO both converge in probability to their optimal values ? ? and F (? ?), respectively: For any ? ¿ 0, we have limt!1 P —?t ? ? — ¿ ? = 0 and limt!1 P —F(t) F (? ?)— ¿ ? = 0. Here is a sketch of the proof of this theorem, the details of which can be found in the supplementary material (Appendix B): 1 ? We focus on {?t }t=1 , which is a stochastic process the filtration of which is defined as Ft = h{y1 , . . . , iyt , yb1 , . . . , ybt }. For this filtration, one can show that b h(?t) is Ft -measurable b and E h(?t)—Ft = h(?t) based on (9). h i 1 b ? As a first step, we can decompose the update rule given in (11) as follows: E bt+1 h(?t) Ft = ? ? 1 1 h(?) + O conditioned on the filtration Ft (see Lemma 7). t bt +2 b2t ? ? P1 ? Next, we show that the sequence 1/bt behaves similarly to 1/t, in the sense that t=1 E 1/b2t ¡ P1 P1 1 1 (see Lemma 8). Moreover, one can show that t=1 E [1/bt] t=1 2t = 1. ? Although h(?) is not differentiable on [0, 1] in general (it can be piecewise linear, for example), one can show that its finite difference is between 1 ?1 and ?1 (see Proposition 9 in the appendix). As a consequence of this result, our process defined in (11) does not get stuck even close to ? ? . ? The ? main part? of the proof is devoted to analyzing the properties of the sequence of t = E (?t ? ?)2 for which we show that limt!1 t = 0, which is sufficient for the statement 5

of the theorem. Our proof follows the convergence analysis of [12]. Nevertheless, our analysis essentially differs from theirs, since in our case, the coefficients cannot be chosen freely. Instead, as explained before, they depend on the labels observed and predicted so far. In addition, the noisy estimation of h(?) depends on the labels, too, but the decomposition step allows us to handle this undesired effect. Remark 4. In principle, the Robbins-Monro algorithm can be applied for finding the root of h(?) as well. This yields an update rule similar to (11), with 1/bt+1 replaced by C/t for a constant C ¿ 0. In this case, however, the convergence of the online F-measure is difficult to analyze (if at all), because the empirical process cannot be written in a nice form. Moreover, as it has been found in the analysis, the coefficient C should be set ? 1/?1 (see Proposition 9 and the choice of {kt } at the end of the proof of Theorem 3). Yet, since ?1 is not known beforehand, it needs to be estimated from the samples, which implies that the coefficients are not independent of the noisy evaluations of h(?)?just like in the case of the OFO algorithm. Interestingly, OFO seems to properly adjust the values 1/bt+1 in an adaptive manner (bt is a sum of two terms, the first of which is t?1 in expectation), which is a very nice property of the algorithm. Empirically, based on synthetic data, we found the performance of the original

Robbins-Monro algorithm to be on par with OFO. As already announced, we are now going to relax the assumption of known posterior probabilities pt = ?(xt). Instead, estimates pbt = gt (xt) ? pt of these probabilities are obtained by classifiers gt that are provided by the external online learner in Algorithm 1. More concretely, assume an online learner A : G ? X ? Y ! G, where G is the set of probabilistic classifiers. Given a current model gt and a new example (xt , yt), this learner produces an updated classifier gt+1 = A(gt , xt , yt). Showing a consistency result for this scenario requires some assumptions on the online learner. With this formal definition of online learner, a statistical consistency result similar to Theorem 3 can be shown. The proof of the following theorem is again deferred to supplementary material (Appendix C). Theorem 5. Assume that the classifiers (gt)1 are provided by an online t=1 in the OFO framework ?R ? learner for which the following holds: There is a ¿ 0 such that E x2X —?(x) gt (x)— d?(x) = O(t

P

P

) . Then, F(t) ! F (? ?) and ?t ! ? ? .

This theorem?s requirement on the online learner is stronger than what is assumed by [11] and recalled in Footnote 2. First, the learner is trained online and not in a batch mode. Second, we also require that the L1 error of the learner goes to 0 with a convergence rate of order t . It might be interesting to note that a universal rate of convergence cannot be established without assuming regularity properties of the data distribution, such as smoothness via absolute continuity. Results of that kind are beyond the scope of this study. Instead, we refer the reader to [5, 6] for details on L1 consistency and its connection to the rate of convergence.

6

Discussion

Regret optimization and stochastic approximation: Stochastic approximation algorithms can be applied for finding the optimum of (4) or, equivalently, to find the unique root of (8) based on noisy evaluations?the latter formulation is better suited for the classic version of the Robbins-Monro root finding algorithm [12]. These algorithms are iterative methods whose analysis focuses on the difference of F (?t) from F (? ?), where ?t denotes the estimate of ? ? in iteration t, whereas our online setting is concerned with the distance of F ((y1 , . . . , yt), (b y1 , . . . , ybt)) from F (? ?), where ybi is the prediction for yi in round i. This difference is crucial because F (?t) only depends on ?t and in addition, if ?t is close to ? ? then F (?t) is also close to F (? ?) (see [19] for concentration properties), whereas in the online F-measure optimization setup, F ((y1 , . . . , yt), (b y1 , . . . , ybt)) can be very different from F (? ?) even if the current estimate ?t is close to ? ? in case the number of previous incorrect predictions is large. In online learning and online optimization it is common to P work with the notion of (cumulative) t regret. In our case, this notion could be interpreted either as i=1 —F ((y1 , . . . , yi), (b y1 , . . . , ybi)) P t F (? ?)— or as i=1 —yi ybi —. After division by t, the former becomes the average accuracy of the F-measure over time and the latter the accuracy of our predictions. The former is hard to interpret because —F ((y1 , . . . , yi), (b y1 , . . . , ybi)) F (? ?)— itself is an aggregate measure of our performance 6

Table 1: Main statistics of the benchmark datasets and one pass F-scores obtained by OFO and 2S methods on various datasets. The bold numbers indicate when the difference is significant between the performance of OFO and 2S methods. The significance level is set to one sigma that is estimated based on the repetitions. Learner: Dataset #instances #pos gisette 7000 3500 news20.bin 19996 9997 Replab 45671 10797 WebspamUni 350000 212189 epsilon 500000 249778 covtype 581012 297711 url 2396130 792145 SUSY 5000000 2287827 kdda 8918054 7614730 kddb 20012498 17244034

#neg #features 3500 5000 9999 1355191 34874 353754 137811 254 250222 2000 283301 54 1603985 3231961 2712173 18 1303324 20216830 2768464 29890095

LogReg OFO 0.954 0.879 0.924 0.912 0.878 0.761 0.962 0.762 0.927 0.934

2S 0.955 0.876 0.923 0.918 0.872 0.762 0.963 0.762 0.926 0.934

Pegasos OFO 0.950 0.879 0.926 0.914 0.884 0.754 0.951 0.754 0.921 0.930

2S 0.935 0.883 0.928 0.910 0.886 0.760 0.950 0.745 0.926 0.929

Perceptron OFO 0.935 0.908 0.914 0.927 0.862 0.732 0.971 0.710 0.913 0.923

2S 0.920 0.930 0.914 0.912 0.872 0.719 0.972 0.720 0.927 0.928

over the first t rounds, which thus makes no sense to aggregate again. The latter, on the other hand, differs qualitatively from our ultimate goal; in fact, —F ((y1 , . . . , yt), (b y1 , . . . , ybt)) F (? ?)— is the alternate measure that we are aiming to optimize for instead of the accuracy.

Online optimization of non-decomposable measures: Online optimization of the F-measure can be seen as a special case of optimizing non-decomposable loss functions as recently considered by [9]. Their framework essentially differs from ours in several points. First, regarding the data generation process, the adversarial setup with oblivious adversary is assumed, unlike our current study where a stochastic setup is assumed. From this point of view, their assumption is more general since the oblivious adversary captures the stochastic setup. Second, the set of classifiers is restricted to differentiable parametric functions, which may not include the F-measure maximizer. Therefore, their proof of vanishing regret does in general not imply convergence to the optimal F-score. Seen from this point of view, their result is weaker than our proof of consistency (i.e., convergence to the optimal F-measure in probability if the posterior estimates originate from a consistent learner). There are some other non-decomposable performance measures which are intensively used in many practical applications. Their optimization had already been investigated in the online or one-pass setup. The most notable such measure might be the area under the ROC curve (AUC) which had been investigated in an online learning framework by [21, 7].

7

Experiments

In this section, the performance of the OFO algorithm is evaluated in a one-pass learning scenario on benchmark datasets, and compared with the performance of the 2-stage F-measure maximization approach (2S) described in Section 2. We also assess the rate of convergence of the OFO algorithm in a pure online learning setup.3 The online learner A in OFO was implemented in different ways, using Logistic Regression (L OG R EG), the classical Perceptron algorithm (P ERCEPTRON) [13] and an online linear SVM called P EGASOS [14]. In the case of L OG R EG, we applied the algorithm introduced in [15] which handles L1 and L2 regularization. The hyperparameters of the methods and the validation procedures are described below and in more detail in Appendix D. If necessary, the raw outputs of the learners were turned into probability estimates, i.e., they were rescaled to [0, 1] using logistic transform. We used in the experiments nine datasets taken from the LibSVM repository of binary classification tasks.4 Many of these datasets are commonly used as benchmarks in information retrieval where the F-score is routinely applied for model selection. In addition, we also used the textual data released in the Replab challenge of identifying relevant tweets [1]. We generated the features used by the winner team [8]. The main statistics of the datasets are summarized in Table 1. 3 4

Additional results of experiments conducted on synthetic data are presented in Appendix F. http://www.csie.ntu.edu.tw/?cjlin/libsvmtools/datasets/binary.html

7

WebspamUni

kdda

url 1 0.9

0.8

0.8

0.8

0.8

0.7

0.7

0.7

0.7

0.6 0.5 0.4

One?pass+LogReg Online+LogReg One?pass+Pegasos Online+Pegasos One?pass+Perceptron Online+Peceptron

0.3 0.2 0.1 0 0 10

2

10

4

10

Num. of samples

6

10

0.6 0.5 0.4

One?pass+LogReg Online+LogReg One?pass+Pegasos Online+Pegasos One?pass+Perceptron Online+Peceptron

0.3 0.2 0.1 0 0 10

1

10

2
10
3
10
4
10
0.6 0.5 0.4
One?pass+LogReg Online+LogReg One?pass+Pegasos Online+Pegasos One?pass+Perceptron Online+Peceptron
0.3 0.2 0.1 0 0 10
5
10
Num. of samples
2
10
4
10
Num. of samples
6
10
(Online) F?score
1 0.9
(Online) F?score
1 0.9
(Online) F?score
(Online) F?score
SUSY 1 0.9
0.6 0.5 0.4
One?pass+LogReg Online+LogReg One?pass+Pegasos Online+Pegasos One?pass+Perceptron Online+Peceptron
0.3 0.2 0.1 0 0 10
2
10
4
10
6
10
Num. of samples

Figure 1: Online F-scores obtained by OFO algorithm on various dataset. The dashed lines represent the one-pass performance of the OFO algorithm from Table 1 which we considered as baseline. One-pass learning. In one-pass learning, the learner is allowed to read the training data only once, whence online learners are commonly used in this setting. We run OFO along with the three classifiers trained on 80% of the data. The learner obtained by OFO is of the form gt?t , where t is the number of training samples. The rest 20% of the data was used to evaluate gt?t in terms of the F-measure. We run every method

on 10 randomly shuffled versions of the data and averaged results. The means of the F-scores computed on the test data are shown in Table 1. As a baseline, we applied the 2S approach. More concretely, we trained the same set of learners on 60% of the data and validated the threshold on 20% by optimizing (6). Since both approaches are consistent, the performance of OFO should be on par with the performance of 2S. This is confirmed by the results, in which significant differences are observed in only 7 of 30 cases. These differences in performance might be explained by the finiteness of the data. The advantage of our approach over 2S is that there is no need of validation and the data needs to be read only once, therefore it can be applied in a pure one-pass learning scenario. The hyperparameters of the learning methods are chosen based on the performance of 2S. We tuned the hyperparameters in a wide range of values which we report in Appendix D. Online learning. The OFO algorithm has also been evaluated in the online learning scenario in terms of the online F-measure (2). The goal of this experiment is to assess the convergence rate of OFO. Since the optimal F-measure is not known for the datasets, we considered the test F-scores reported in Table 1. The results are plotted in Figure 1 for four benchmark datasets (the plots for the remaining datasets can be found in Appendix G). As can be seen, the online F-score converges to the test F-score obtained in one-pass evalaution in almost every case. There are some exceptions in the case of P EGASOS and P ERCEPTRON. This might be explained by the fact that SVM-based methods as well as the P ERCEPTRON tend to produce poor probability estimates in general (which is a main motivation for calibration methods turning output scores into valid probabilities [3]).

8

Conclusion and Future Work

This paper studied the problem of online F-measure optimization. Compared to many conventional online learning tasks, this is a specifically challenging problem, mainly because of the nondecomposable nature of the F-measure. We presented a simple algorithm that converges to the optimal F-score when the posterior estimates are provided by a sequence of classifiers whose L1 error converges to zero as fast as t for some ¿ 0. As a key feature of our algorithm, we note that it is a purely online approach; moreover, unlike approaches such as 2S, there is no need for a hold-out validation set in batch mode. Our promising results from extensive experiments validate the empirical efficacy of our algorithm. For future work, we plan to extend our online optimization algorithm to a broader family of complex performance measures which can be expressed as ratios of linear combinations of true positive, false positive, false negative and true negative rates [10]; the F-measure also belongs to this family. Moreover, going beyond consistency, we plan to analyze the rate of convergence of our OFO algorithm. This might be doable thanks to several nice properties of the function h(?). Finally, an intriguing question is what can be said about the case when some bias is introduced because the classifier gt does not converge to ?. Acknowledgments. Krzysztof Dembczy?nski is supported by the Polish National Science Centre under grant no. 2013/09/D/ST6/03917. 8

2 References

[1] E. Amig?o, J. C. de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Mart??n-Wanton, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2013: Evaluating online reputation monitoring systems. In CLEF, volume 8138, pages 333?352, 2013. [2] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends in Machine Learning, 5(1):1?122, 2012. ? o, and Gy. Szarvas. Tune and mix: Learning to rank using [3] R. Busa-Fekete, B. K?egl, T. Eltet? ensembles of calibrated multi-class classifiers. Machine Learning, 93(2?3):261?292, 2013. [4] N. Cesa-Bianchi and G. Lugosi. Prediction, Learning, and Games. Cambridge University Press, 2006. [5] L. Devroye and L. Gy?orfi. Nonparametric Density Estimation: The L1 View. Wiley, NY, 1985. [6] L. Devroye, L. Gy?orfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, NY, 1996. [7] W. Gao, R. Jin, S. Zhu, and Z.-H. Zhou. One-pass AUC optimization. In ICML, volume 30:3, pages 906?914, 2013. [8] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In Working Notes of CLEF 2013 Evaluation Labs and Workshop, 2013. [9] P. Kar, H. Narasimhan, and P. Jain. Online and stochastic gradient methods for nondecomposable loss functions. In NIPS, 2014. [10] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In NIPS, pages 2744?2752, 2014. [11] H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In NIPS, 2014. [12] H. Robbins and S. Monro. A stochastic approximation method. Ann. Math. Statist., 22(3):400? 407, 1951. [13] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386?408, 1958. [14] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In ICML, pages 807?814, 2007. [15] Y. Tsuruoka, J. Tsujii, and S. Ananiadou. Stochastic gradient descent training for L1regularized log-linear models with cumulative penalty. In ACL, pages 477?485, 2009. [16] C.J. van Rijsbergen. Foundation and evalaution. Journal of Documentation, 30(4):365?373, 1974. [17] S. R. S. Varadhan. Probability Theory. New York University, 2000. [18] W. Waegeman, K. Dembczy?nski, A. Jachnik, W. Cheng, and E. H?ullermeier. On the Bayesoptimality of F-measure maximizers. Journal of Machine Learning Research, 15(1):3333? 3388, 2014. [19] N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu. Optimizing F-measure: A tale of two approaches. In ICML, 2012. [20] M. Zhao, N. Edakunni, A. Pocock, and G. Brown. Beyond Fano?s inequality: Bounds on the optimal F-score, BER, and cost-sensitive risk and their implications. JMLR, pages 1033?1090, 2013. [21] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang. Online AUC maximization. In ICML, pages 233?240, 2011.

9