From Cost-Sensitive Classification to Tight F-measure Bounds

Kevin Bascol
Laboratoire Hubert Curien
UMR 5516, Univ Lyon, UJM
F-42023, Saint-Etienne, France
Bluecime inc., France

Rémi Emonet
Laboratoire Hubert Curien
UMR 5516, Univ Lyon, UJM
F-42023, Saint-Etienne, France

Elisa Fromont
IRISA/Inria,
Univ. Rennes 1,
35042 Rennes cedex, France

Amaury Habrard
Laboratoire Hubert Curien
UMR 5516, Univ Lyon, UJM
F-42023, Saint-Etienne, France

Guillaume Metzler
Laboratoire Hubert Curien
UMR 5516, Univ Lyon, UJM
F-42023, Saint-Etienne, France
Blitz inc., France

Marc Sebban
Laboratoire Hubert Curien
UMR 5516, Univ Lyon, UJM
F-42023, Saint-Etienne, France

Abstract

The F-measure is a classification performance measure, especially suited when dealing with imbalanced datasets, which provides a compromise between the precision and the recall of a classifier. As this measure is non convex and non linear, it is often indirectly optimized using cost-sensitive learning (that affects different costs to false positives and false negatives). In this article, we derive theoretical guarantees that give tight bounds on the best F-measure that can be obtained from cost-sensitive learning. We also give an original geometric interpretation of the bounds that serves as an inspiration for **CONE**, a new algorithm to optimize for the F-measure. Using 10 datasets exhibiting varied class imbalance, we illustrate that our bounds are much tighter than previous work and show that **CONE** learns models with either superior F-measures than existing methods or comparable but in fewer iterations.

1 Introduction

The F-measure (van Rijsbergen, 1974) is a performance measure used in classification to evaluate the ability of a classifier to predict the labels of new instances with a good recall and a good precision (as an harmonic mean of these two measures). It is the most commonly used measure in imbalanced settings where using the accuracy of the classifier would greatly favor the majority class (Chandola et al., 2009; Lopez et al., 2013). This measure is parameterized by a parameter β that controls the relative importance of the precision and the recall. For $\beta < 1$ (resp. $\beta > 1$), more importance is given to the precision (resp. recall), with $\beta = 1$, they are considered equally important. The F-measure can be expressed in terms of the true positive rate and true negative rate of the model. These rates are count-based measures which makes the F-measure, in addition to being non convex, unsuitable for direct optimization (Narasimhan et al., 2015a).

Several methods have been studied to solve the F_β-measure optimization problem. They can roughly be separated into two categories: *Decision Theoretic Approaches (DTA)* (Dembczyński et al., 2017) which tries to find the classifier that maximizes the expectation of the F-measure. More precisely, these methods usually fit a probabilistic model during training followed by an inference procedure at prediction time (Decubber et al., 2018). The probabilistic model can be learned by optimizing a “simpler” surrogate function (e.g., (Dembczynski et al., 2011; Jansche, 2005; Ye et al., 2012; P.M. Chinta and Murty, 2013)). The second category consists of *Empirical Utility Maximization (EUM)* methods that learn multiple accurate models with different parameters and keep the model which maximizes the F-measure (Busa-Fekete et al., 2015; Joachims, 2005; Musicant et al., 2003; Parambath et al., 2014; Zhao et al., 2013; Narasimhan et al., 2015b). In this second category, the parameters can be the different

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

classification thresholds for probabilistic models (Busa-Fekete et al., 2015; Joachims, 2005; Zhao et al., 2013; Narasimhan et al., 2015b) or the costs on the classification errors for cost-sensitive methods (Musicant et al., 2003; Parambath et al., 2014; Koyejo et al., 2014). *EUM* methods focus on *estimation* on a possibly infinite training set while *DTA* approaches are concerned with *generalization* performance (Dembczyński et al., 2017). Ye et al. (2012) shows that both categories of methods give asymptotically the same results and propose heuristics to decide on the category to use depending on the context.

The work presented in this paper falls into the *EUM* based methods within a cost-sensitive classification approach. Indeed, by taking into account some per-class misclassification-costs, cost-sensitive learning aims at dealing with problems induced by class-imbalanced datasets. One of the few recent papers addressing the F-measure optimization, from a theoretical point of view, (see also (Busa-Fekete et al., 2015; Zhao et al., 2013; Koyejo et al., 2014; Narasimhan et al., 2015b)) is the work from (Parambath et al., 2014) . The authors propose a grid-based approach to find the optimal costs for which a cost-sensitive classifier would give the best F-measure. They theoretically prove that, with a sufficiently precise grid, one can be arbitrarily close to the optimal F-measure. However, this method relies on a relatively loose result which imposes to parse the whole grid leading an unnecessary computational burden. The methods proposed by Koyejo et al. (2014) and by Narasimhan et al. (2015b) achieve good performances with a limited time budget using a cost-sensitive approach. They roughly consist of fitting a probabilistic model then using a threshold in order to optimize the F-measure. In the first cases the threshold is tuned on a validation set while an iterative process based on the bisection algorithm (Boyd and Vandenberghe, 2004). However we will see that, despite their simplicity (and the theoretical guarantees provided), it is possible to achieve higher performance by training (a few) number of models. Indeed tuning a model is not enough and we need to learn a different hyperplane to take the costs on each class into account. In this article, we propose a novel tighter theoretical result for cost-sensitive-based algorithms which allows us to derive a new efficient algorithm for F-measure optimization. Our contributions can be summarized as follows:

- we demonstrate tight theoretical guarantees on the F-measure of classifiers obtained from cost-sensitive learning;
- we give a geometric interpretation of the theoretical guarantees: they can be represented as unreachable regions (cones) in a 2D space where the x-axis gives the value of a parameter t that controls the relative costs of the considered classes, and the y-axis gives the F-measure of the corresponding cost-sensitive classifier;
- going beyond traditional asymptotic analysis, we study the actual behavior of our bounds, on real datasets, showing it is much tighter than previous existing results;
- inspired by our bounds and their interpretation, we introduce an algorithm to explore the space of costs: our experiments show the relevance of (i) using our algorithm compared to other baselines (such as Parambath et al. (2014)), and (ii) retraining the model iteratively compared to the previously described methods (Koyejo et al. (2014); Narasimhan et al. (2015b)) that only tune an offset or threshold.

In Section 2, we introduce the notations and present our theoretical bound on the optimal F-measure based on a cost-sensitive approach and the pseudo-linear property of the F-measure. We give a geometric interpretation in Section 3 and introduce an algorithm that iteratively selects classification costs that lead to a near-optimal F-measure. Section 4 is devoted to the experiments on real datasets. These experiments show the effectiveness of the proposed bounds from a practical point of view. Furthermore, they show that it is possible to reach higher performance than a single model tuned *a posteriori* or much faster than grid search methods. We finally conclude in Section 5.

2 Theoretical Bounds

Because of limited space, the following proofs focus on the binary classification case, the multi-class results are given in the supplementary material.

2.1 Notations

Let $\mathbf{X} = (x_1, ..., x_m)$, where $\boldsymbol{x}_i \in \mathbb{R}^n$, be the set of m training instances and $\mathbf{Y} = (y_1, ..., y_m)$ their corresponding label, where $y \in \{0, 1\}$. Let $\mathcal{H}$ be a family of hypothesis e.g., linear separators. For a given hypothesis $h \in \mathcal{H}$ learned from $(\mathbf{X}, \mathbf{Y})$, the errors that h makes can be summarized in an error profile, noted $\mathbf{E}(h)$, which, in the binary case can be defined as $(\mathtt{FN}(h), \mathtt{FP}(h))$.

In a binary setting, P is the proportion of positive instances and N the proportion of negative examples. We also denote by $\boldsymbol{e}$ the vector (e_1, e_2) where e_1 and e_2 are respectively the proportion of False Negative (FN) examples and the proportion of False Positive

(FP) ones as introduced previously. We then denote as $\mathcal{E}(\mathcal{H})$ the set of all possible error profiles for a given set of hypotheses: an error profile $\boldsymbol{e} = (e_1, e_2)$ is in $\mathcal{E}(\mathcal{H})$ if there exists an hypothesis $h \in \mathcal{H}$ that yields proportions of e_1 false negatives and e_2 false positives.

We first recall the definition of F-measure for any value of β:

$$F_\beta = \frac{(1+\beta^2)(P - FN)}{(1+\beta^2)P - FN + FP}.$$

Using the above notations, the F-measure, $F_\beta(\boldsymbol{e})$, defined in terms of the error profile $\boldsymbol{e}$ can be rewritten as:

$$F_\beta(\boldsymbol{e}) = \frac{(1+\beta^2)(P - e_1)}{(1+\beta^2)P - e_1 + e_2}.$$

2.2 Pseudo linearity property

The F-measure is a linear-fractional function, i.e. it can be written as the ratio of two affine functions of the error profile. We briefly recall how to show that the F-measure is a pseudo-linear function, which is one of the main property of linear-fractional function. This property is the starting point of the demonstration of our main theoretical result.

Definition 1. *[from Rapcsák (1991)] A real differentiable function f defined on an open convex set $\mathcal{C} \subset \mathbb{R}^q$ is said to be pseudo-convex if for every $\boldsymbol{e}, \boldsymbol{e}' \in \mathcal{C}$,*

$$\langle \nabla f(\boldsymbol{e}), (\boldsymbol{e}' - \boldsymbol{e}) \rangle \geq 0 \implies f(\boldsymbol{e}') \geq f(\boldsymbol{e}),$$

where ∇f denotes the gradient of the function f.

The pseudo-convexity is used to define the pseudo-linearity as we see below.

Definition 2. *A function f defined on an open convex $\mathcal{C}$ is said to be pseudo-linear if both f and $-f$ are pseudo-convex.*

It is now easy to show that the F-measure has the property of pseudo-linearity.

Proposition 1. *The F-measure is a pseudo-linear function.*

Proof 1. *See Supplementary Material.*

Using this property, we are able, using a result from Alberto and Laura (2009) to give a link between the F-measure and a cost-sensitive function, i.e. a function which assigns weights to each classes.

Proposition 2. *[Theorem 3.3.9 from Alberto and Laura (2009)] Let f be a non-constant differentiable function on an open convex set $\mathcal{C} \in \mathbb{R}^q, q > 0$. Then f is pseudo-linear on $\mathcal{C}$ if and only if the following properties hold:*
(i) each of the level sets of f is the intersection of $\mathcal{C}$ with a hyperplane;
(ii) $\nabla f(\boldsymbol{e}) \neq 0$ for all $\boldsymbol{e} \in \mathcal{C}$.

Let us consider the set of error profile $\{\boldsymbol{e} \in \mathbb{R}^2 \mid (1+\beta^2)P - e_1 + e_2 > 0\}$ (which is always the case in practice with the F-measure). Then according to the previous theorem, we rewrite (i) as follows:
It exists $\boldsymbol{a} : \mathbb{R} \to \mathbb{R}^2$ and $b : \mathbb{R} \to \mathbb{R}$ such that

$$F(\boldsymbol{e}) = t \iff \langle \boldsymbol{a}(t), \boldsymbol{e} \rangle + b(t) = 0,$$

which can be rewritten :

$$\langle \boldsymbol{a}(F(\boldsymbol{e})), \boldsymbol{e} \rangle + b(F(\boldsymbol{e})) \quad = \quad 0. \qquad (1)$$

For the F-measure, the functions $\boldsymbol{a}$ and b are defined by $\boldsymbol{a}(t) = (1+\beta^2-t, t)$ and $b(t) = (1+\beta^2)P(t-1)$. The term $\langle \boldsymbol{a}(t), \boldsymbol{e} \rangle$ can be seen as a weighted error loss function, and thus $\boldsymbol{a}(t)$ can be seen as the costs to assign to each class.

2.3 Bounds on the optimal F-measure

We now show the importance of the function $\boldsymbol{a}$ and of the parameter t to characterize the difference of F-measure between any two error profiles.

Step 1: impact of a change in the error profile

We first derive the relation between the difference in F-measure (F) and the difference in error profile $(\boldsymbol{e})$. We thus consider $\boldsymbol{e}$ and $\boldsymbol{e}'$ any two error profiles and denote by $F(\boldsymbol{e})$ and $F(\boldsymbol{e}')$ the corresponding F-measures.

From the pseudo-linearity property (Eq. (1)), we have:

$$\begin{aligned} 0 &= \langle \boldsymbol{a}(F(\boldsymbol{e})), \boldsymbol{e} \rangle + b(F(\boldsymbol{e})), && (2) \\ 0 &= \langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e}' \rangle + b(F(\boldsymbol{e}')). && (3) \end{aligned}$$

We now develop $\langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e} - \boldsymbol{e}' \rangle$ and make the difference in F-measure appears in its expression.

$$\begin{aligned} \langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e} - \boldsymbol{e}' \rangle &= \langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e} \rangle + b(F(\boldsymbol{e}')), \\ &= \langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e} \rangle - \langle \boldsymbol{a}(F(\boldsymbol{e})), \boldsymbol{e} \rangle \\ &\quad -b(F(\boldsymbol{e})) + b(F(\boldsymbol{e}')), \\ \langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e} - \boldsymbol{e}' \rangle &= (F(\boldsymbol{e}') - F(\boldsymbol{e})) \\ &\quad \cdot \left((1+\beta^2)P_1 - e_1 + e_2\right), \end{aligned}$$

where the first line uses the linearity of the inner product and Eq. (3). The second uses Eq. (2) and the last line uses the definition of $\boldsymbol{a}$ and b defined in Section 2.2.

Now we can rewrite the difference in F-measure as:

$$F(\boldsymbol{e}') - F(\boldsymbol{e}) \quad = \quad \Phi_{\boldsymbol{e}} \cdot \langle \boldsymbol{a}(F(\boldsymbol{e}')), \boldsymbol{e} - \boldsymbol{e}' \rangle, \qquad (4)$$

where $\Phi_{\boldsymbol{e}} = \dfrac{1}{(1+\beta^2)P - e_1 + e_2}$.

Step 2: bounds on the F-measure $F(\boldsymbol{e}')$

We suppose that we have a value of t for which a weighted-classifier with weights $\boldsymbol{a}(t)$ has been learned. This classifier has an error profile $\boldsymbol{e}$ and a F-measure $F(\boldsymbol{e})$. We now imagine a hypothetical classifier that is learned with weights $\boldsymbol{a}(t')$, and we denote by $\boldsymbol{e}'$ the error profile of this classifier. For any value of t', we derive an upper bound on the on the F-measure $F(\boldsymbol{e}')$ that this hypothetical classifier can achieve.

Starting from the result obtained in Eq. 4, we have:

$$\begin{aligned}
& F(\boldsymbol{e}') - F(\boldsymbol{e}) \\
= \;& \Phi_{\boldsymbol{e}} \left(\langle \boldsymbol{a}(t'), \boldsymbol{e} \rangle - \langle \boldsymbol{a}(t'), \boldsymbol{e}' \rangle \right), \\
= \;& \Phi_{\boldsymbol{e}} \left(\langle \boldsymbol{a}(t), \boldsymbol{e} \rangle + \langle \boldsymbol{a}(t') - \boldsymbol{a}(t), \boldsymbol{e} \rangle - \langle \boldsymbol{a}(t'), \boldsymbol{e}' \rangle \right), \\
= \;& \Phi_{\boldsymbol{e}} \left(\langle \boldsymbol{a}(t), \boldsymbol{e} \rangle + (t'-t)(e_2 - e_1) - \langle \boldsymbol{a}(t'), \boldsymbol{e}' \rangle \right), \\
\leq \;& \Phi_{\boldsymbol{e}} \left(\langle \boldsymbol{a}(t), \boldsymbol{e}' \rangle + \varepsilon_1 - \langle \boldsymbol{a}(t'), \boldsymbol{e}' \rangle + (t'-t)(e_2 - e_1) \right), \\
\leq \;& \Phi_{\boldsymbol{e}} \left((t-t')(e_2' - e_1') + \varepsilon_1 + (t'-t)(e_2 - e_1) \right), \\
\leq \;& \Phi_{\boldsymbol{e}} \varepsilon_1 + \Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - (e_2' - e_1'))(t'-t).
\end{aligned}$$

We have successively used the linearity of the inner product, introduced $\boldsymbol{a}(t)$ and its definition in the first three equalities. The first inequality uses $\langle \boldsymbol{a}(t), \boldsymbol{e} \rangle \leq \langle \boldsymbol{a}(t), \boldsymbol{e}_{best} \rangle + \varepsilon_1$, the sub-optimality of the $\boldsymbol{a}(t)$-weighted-error classifier. The value of ε_1 represents the excess of risk of the classifier which aim to minimize the $\boldsymbol{a}(t)$-weighted-error. More precisely, it represents the difference of risk between our classifier and the best classifier h_{best} (in terms on $\boldsymbol{a}(t)$-weight-error) in our set of hypothesis $\mathcal{H}$. We denote by $\boldsymbol{e}_{best}$ the error profile associated to h_{best}. This inequality is still true if we replace $\boldsymbol{e}_{best}$ by any vector $\boldsymbol{e}'$. We finally apply the definition of $\boldsymbol{a}$.
The quantities e_2' and e_1' remain unknown, but can be tightly bounded. The result of this development can be summarized into the following proposition (see the supplementary material for the derivation of the values of M_{min} and M_{max}).

Proposition 1. *Let* $\boldsymbol{e}$ *be the error profile obtained with a classifier trained with the parameter* t *and* $F(\boldsymbol{e})$ *its associated F-measure value. Let us also consider* $\Phi_{\boldsymbol{e}}$ *as defined in Eq.* (4) *and* $\varepsilon_1 > 0$ *the sub-optimality of our linear classifier. Then for all* $t' < t$:

$$F(\boldsymbol{e}') \leq F(\boldsymbol{e}) + \Phi_{\boldsymbol{e}} \varepsilon_1 + \Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - M_{max})(t'-t),$$

where $M_{max} = \underset{\substack{\boldsymbol{e}'' \in \mathcal{E}(\mathcal{H}) \\ s.t.\ F(\boldsymbol{e}'') > F(\boldsymbol{e})}}{max} \quad (e_2'' - e_1'')$
and, for all $t' > t$:

$$F(\boldsymbol{e}') \leq F(\boldsymbol{e}) + \Phi_{\boldsymbol{e}} \varepsilon_1 + \Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - M_{min})(t'-t),$$

where $M_{min} = \underset{\substack{\boldsymbol{e}'' \in \mathcal{E}(\mathcal{H}) \\ s.t.\ F(\boldsymbol{e}'') > F(\boldsymbol{e})}}{min} \quad (e_2'' - e_1'')$.

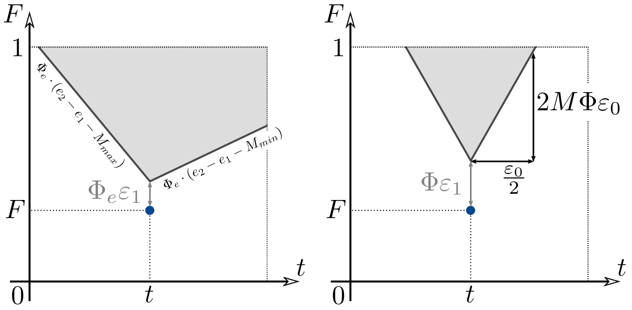
F
1
$\Phi_e \cdot (e_2 - e_1 - M_{max})$
$\Phi_e \cdot (e_2 - e_1 - M_{min})$
$\Phi_e \varepsilon_1$
F
0
t
t
F
1
$2M\Phi\varepsilon_0$
$\frac{\varepsilon_0}{2}$
$\Phi\varepsilon_1$
F
0
t
t

Figure 1: Geometric interpretation of both theoretical results: our bound on the left and the one from Parambath et al. (2014) on the right. Note that our "cone" is not symmetric compared to the other one. On the left, the slanted values represent the slope of our cone on each side : $\Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - M_{max})$ and $\Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - M_{min})$.

With this first result, we give an upper bound on the reachable F-measures for any value of t' given an observed value of F-measure with the parameter t. A geometric interpretation and an illustration of this result will be provided in Section 3.1.

Corollary 1. *Given the same assumptions and considering* $t^\star$ *the value of* t *for which the best cost-sensitive learning algorithm leads to a model with an error profile* $e^\star$ *associated to the optimal F-measure, we have: if* $t^\star < t$:

$$F(\boldsymbol{e}^\star) \leq F(\boldsymbol{e}) + \Phi_{\boldsymbol{e}} \varepsilon_1 + \Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - M_{max})(t^\star - t),$$

and, if $t^\star > t$:

$$F(\boldsymbol{e}^\star) \leq F(\boldsymbol{e}) + \Phi_{\boldsymbol{e}} \varepsilon_1 + \Phi_{\boldsymbol{e}} \cdot (e_2 - e_1 - M_{min})(t^\star - t).$$

This means that if we learn a model with a parameter t sufficiently close to $t^\star$ then, we guarantee to reach the optimal F-measure up to a constant equal to $\Phi_{\boldsymbol{e}} \varepsilon_1$.

3 Geometric Interpretation, CONE

In this section we provide a geometric interpretation of our main result, i.e. Proposition 1 of Section 2.3 and compare it to the bound introduced in Parambath et al. (2014). We also show how this theoretical result can be an inspiration to create an algorithm, **CONE**, which optimizes the F-measure by wrapping a cost-sensitive learning algorithm.

3.1 Unreachable regions

In Fig. 1 (left), we give a geometric interpretation of the result from Prop. 1 in the 2-D space where t is the x-axis and F is the y-axis. In this (t, F) graph, the previous near-optimality result yields an upper cone of values where $F(\boldsymbol{e}^\star)$ cannot be found. More precisely, when a model is learned for a given value of t (with weights $\boldsymbol{a}(t)$), we measure the value $F(\boldsymbol{e})$ of this model and, given these two numbers, we are able to draw an upper cone which represents the unreachable values of F-measure for any t' on the x-axis. Furthermore, given ε_1, the sub-optimality of the cost-sensitive learning algorithm for the weighted-0/1 loss, $\Phi_{\boldsymbol{e}}\varepsilon_1$ corresponds to the vertical offset of this cone, which means that the peak of the cone is located at $(t, F(\boldsymbol{e}) + \Phi_{\boldsymbol{e}}\varepsilon_1)$.

Note that, even if the authors were focusing on asymptotic results, the bound given in Parambath et al. (2014) can also be interpreted geometrically. Their bound is given as follows:

$$F(\boldsymbol{e}^\star) \leq F(\boldsymbol{e}) + \Phi \cdot (2\varepsilon_0 M + \varepsilon_1),$$

where $M = \underset{\boldsymbol{e}\in\mathcal{E}(\mathcal{H})}{max} \|\boldsymbol{e}\|_2$, $\Phi = (\beta^2 P)^{-1}$ and ε_0 is a gap parameter defined as the ℓ_2 norm of the difference between a weighted function $\boldsymbol{a}$ and the optimal one $\boldsymbol{a}^\star$. In the supplementary material, we detail how this bound can, in fact, be rewritten for all $t, t' \in [0,1]$ as:

$$F(\boldsymbol{e}') \leq F(\boldsymbol{e}) + \Phi\varepsilon_1 + 4M\Phi|t' - t|.$$

This bound also defines a cone which is, this time, symmetric with a slope equal to $4\Phi M$, as illustrated in Fig. 1 (right). Using real datasets, Section 4.2 compares the cones produced by this bound and ours.

3.2 A bound-inspired algorithm

We now leverage the geometric interpretation from Section 3.1 to design **CONE** (Cone-based Optimal Next Evaluation), an iterative algorithm that wraps a cost-sensitive classification algorithm (e.g., a weighted SVM). At every iteration i, **CONE** proposes a new value t_i to be used by the cost-sensitive algorithm. **CONE** is illustrated in Fig. 3 and is explained below.

The choice of t_i is based on the area $\mathcal{Z}_{i-1}$ which we define as the union of all cones obtained from previous iterations. t_i is chosen to reduce the maximum value of F for which (t, F) is not in any previous cone. To achieve this goal, **CONE** keeps track of a list L, initialized with the values 0 and 1, and enriched at each iteration with the values of t that have been considered. The selection of t_i is done as follows: (i) search the value t_{opt} which maximizes $F_{max}(t) = \max\{F, (t, F) \notin \mathcal{Z}_{i-1}\}$, (ii) search for the greatest value t_l in L such that $t_l < t_{opt}$ and the smallest value t_r such that $t_{opt} < t_r$. (iii) take the middle of the interval $[t_l, t_r]$ as the return value, i.e. $t_i = \frac{1}{2}(t_l + t_r)$.

The cost sensitive classification algorithm then provides a new value of F_i obtained from cost t_i, which is used to refine the unreachable area as $\mathcal{Z}_i = \mathcal{Z}_{i-1} \cup \mathcal{V}_i$, where $\mathcal{V}_i$ is the cone corresponding to (t_i, F_i). In the case where there are multiple values of t that maximize $F_{max}(t)$ (e.g., at the beginning, or when some range of t values yield $F = 1$), **CONE** selects as t_{opt} the middle of the widest range at the first stage (i) (see the white dotted lines in Fig. 3).

```
Input: training set $S$,
Input: weighted-learning algorithm $wLearn$,
Input: stopping criterion $shouldStop$.
Initialize $L = \{0, 1\}$, $\mathcal{Z}_0 = \varnothing$ and $i = 1$.
repeat
    $t_i = findNextT(\mathcal{Z}_{i-1}, L)$
    $classifier_i = wLearn(1 + \beta^2 - t_i, t_i, S)$
    $F_i = F_\beta(classifier_i, S)$
    $\mathcal{V}_i = unreachableZone(t_i, F_i, S, classifier_i)$
    $\mathcal{Z}_i = \mathcal{Z}_{i-1} \cup \mathcal{V}_i$
    $L = L \cup \{t_i\}$
    $i = i + 1$
until $shouldStop(i, classifier_i, \mathcal{Z}_i, L)$
```

Figure 2: **CONE** Algorithm

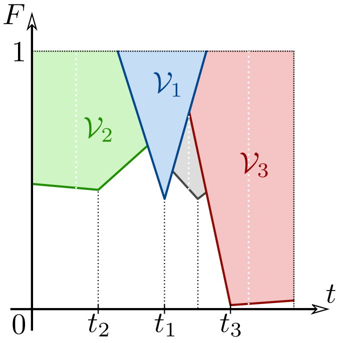

Figure 3: Illustration of the **CONE** algorithm in the middle of its fourth iteration. The colored areas represent the unreachable regions in the (t, F)-space.

From a practical perspective, $\mathcal{Z}_i$ can be represented as a combination of linear constraints or as a very dense grid of binary values (a rasterization of $[0,1] \times [0,1]$, the (t, F) space). Both approaches can be made efficient (and negligible compared to $wLearn$). The stopping criterion $shouldStop$ can take different forms including a fixed number of iterations, a fixed time budget, or some rules on the current best F-measure and the current upper bound $max_t\, F_{max}(t)$. While the algorithm we describe selects a single next value of t to consider, it can easily be generalized to produce multiple values of t to consider in parallel (to exploit parallel computing of multiple instances of $wLearn$).

By always selecting a t_i that is in the middle of two

Table 1: Datasets details. The Imbalance Ratio (I.R.) is the ratio between negative and positive instances (or between sizes of the largest and smallest classes, in a multiclass setting).

Dataset	Instances	Classes	I.R.	Features
Adult	48842	2	3.19	123
Abalone10	4174	2	5.64	10
SatImage	6400	2	9.3	36
IJCNN'01	141691	2	9.39	22
Abalone12	4174	2	15.18	10
PageBlocks	5500	2	22.7	10
Yeast	1484	2	27.48	8
Wine	1599	2	28.79	11
Letter	20000	26	1.32	16
News20	19928	20	1.12	62061

previously tested t-values, **CONE** performs a progressive refinement of a grid. We can (and do, in practice) restrict the values of t in the (t, F)-space that the algorithm considers. More precisely, we can limit the depth of the progressive refinement to an integer value k. In this case, and **CONE** will do at most $2^k - 1$ iterations, in order to cover all possible values on a grid with stride $\frac{1}{2^k}$. However, as the procedure is informed by the theoretical bounds, we will see in Section 4.3 that **CONE** finds good models in its very first iterations.

4 Experiments

The experiments from this section study the tightness of our bounds and behavior of the **CONE** algorithm.

4.1 Datasets and experimental settings

Table 1 describes the datasets we used for our experiments, with their Imbalance Ratio (I.R.). The higher this ratio, the more one should expect that optimizing the classification accuracy is a bad choice in terms of trade-off between precision and recall. The datasets *IJCNN'01* and *News20* are obtained from LIBSVM[1]. The other ones come from the UCI repository[2].

We reproduce the experimental settings from Parambath et al. (2014) which we describe here. For datasets with no explicit test set, $\frac{1}{4}$ of the data is kept for testing. The training set is split at random, keeping $\frac{1}{3}$ as the validation set, used to select the hyper-parameters using the F_1-measure. The penalty constraint of the classifiers (hyper-parameter C) is considered in $\{2^{-6}, 2^{-5}, ..., 2^{6}\}$. In the experiments t is taken in $[0, 1]$ as t belongs in the image space of the F-measure. Thus the class weights $\boldsymbol{a}(t)$ belongs to $[0, 1 + \beta^2]$. The maximal number of training iterations is set to 50000. Fitting the intercept of the classifiers is achieved by adding a constant feature with value 1. We report test-time F_1-measure averaged over 5 experiments.

We consider two different base cost-sensitive classification algorithms (both implementations use LIBLINEAR): linear SVM and Logistic Regression (LR) for a fair comparison with Koyejo et al. (2014). We report the performance of 5 different approaches: using a single standard classification algorithm with hyper-parameters tuned on the F-measure, the **Grid** wrapper proposed in Parambath et al. (2014) that regularly splits the interval $[0, 1]$ of t values, the algorithm derived from our theoretical study, algorithm 2 from Narasimhan et al. (2015b) based on the bisection method, and finally, an additional baseline (with the $_{I.R.}$ subscript), which consists in using a cost that re-balances the classes (the cost c of a False Negative is the proportion of positive examples in the dataset and the cost of False Positive is $1 - c$).

About ε_1. The value of ε_1 (in all presented bounds) represents the $\boldsymbol{a}(t)$-weighted sub-optimality of the classifier, compared to the best one from the hypothesis class. This sub-optimality cannot be computed efficiently as it would require a learning algorithm that produces optimal classifiers in terms of $\boldsymbol{a}(t)$-weighted error. We thus start by studying the impact of ε_1 in Section 4.2 on our bounds. As the focus of this paper is not on estimating ε_1, we then set $\varepsilon_1 = 0$ which is computationnaly free, and shown by the experiment to be a reasonable choice both in terms of bound analysis (the bound is most of the time respected) and in terms of overall results from the **CONE** algorithm.

4.2 Evaluation of the tightness of the bound

In this section, we aim at illustrating and showing the tightness of our bounds. To do so, we consider the (t, F) values obtained by 19 weighted-SVM learned on a regular grid of t values. For these same 19 models, we consider the cones obtained from our bounds and previous work (see Section 3.1 for details). Due to space limitations, we show only two illustrations, with two different datasets, but the supplementary material contains similar illustrations for all datasets.

Impact of ε_1. Both our bounds and the one from previous work are impacted by ε_1 which shows up as an offset, multiplied by $\Phi_{\boldsymbol{e}}$ for our bounds, and by Φ in previous work. As $\Phi_{\boldsymbol{e}} \leq \Phi$, our bounds are less impacted by an increased ε_1. With the 19-SVM setting, Fig. 4 shows the evolution of the maximum still-

[1] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[2] `https://archive.ics.uci.edu/ml/datasets.html`

Table 2: Classification F-Measures for $\beta = 1$ with SVM and Logistic Regression algorithms. SVM$_G$ and LR$_G^T$ are reproduced experiments of Parambath et al. (2014) and the subscript $_{I.R.}$ is used for the classifiers trained with a cost depending on the Imbalance Ratio. The subscript $_B$ corresponds to the bisection algorithm presented by Narasimhan et al. (2015b). LR^T and LR$^T_{I.R.}$ are reproduced experiments of Koyejo et al. (2014). Finally the $_C$ stands for our wrapper **CONE** and SVM$_C^T$ designed as a combination using the **CONE** + threshold. Reported F-measure values are averaged over 5 experiments (standard deviation between brackets).

Dataset	SVM	SVM$_{I.R.}$	SVM$_G$	SVM$_C$	SVM$_C^T$	LR^T	LR$^T_{I.R.}$	LR^T_G	LR$_B$
Adult	62.5 (0.2)	64.9 (0.3)	66.4 (0.1)	66.5 (0.1)	66.4 (0.1)	66.5 (0.1)	66.5 (0.1)	66.5 (0.1)	66.6 (0.1)
Abalone10	0.0 (0.0)	30.9 (1.2)	32.4 (1.3)	32.2 (0.8)	31.8 (1.9)	30.8 (2.2)	30.7 (1.9)	30.7 (1.9)	31.6 (0.6)
Satimage	0.0 (0.0)	23.4 (4.3)	20.4 (5.3)	20.6 (5.6)	30.9 (2.0)	21.2 (11.1)	28.6 (1.9)	28.6 (1.9)	21.4 (4.6)
IJCNN	44.5 (0.4)	53.3 (0.4)	61.6 (0.6)	61.6 (0.6)	62.6 (0.4)	59.4 (0.5)	56.5 (0.3)	56.5 (0.3)	59.2 (0.3)
Abalone12	0.0 (0.0)	16.8 (2.7)	16.8 (4.2)	18.3 (3.3)	16.3 (3.0)	15.5 (3.1)	17.0 (3.3)	17.0 (3.3)	17.7 (3.7)
Pageblocks	48.1 (5.8)	39.6 (4.7)	66.4 (3.2)	62.8 (3.9)	67.6 (4.0)	59.2 (8.1)	55.9 (6.4)	55.9 (6.4)	55.7 (5.7)
Yeast	0.0 (0.0)	29.4 (2.9)	38.6 (7.1)	39.0 (7.5)	35.4 (15.6)	37.4 (10.1)	39.9 (6.5)	27.6 (6.8)	27.6 (6.8)
Wine	0.0 (0.0)	15.6 (5.2)	20.0 (6.4)	22.7 (6.0)	19.3 (7.9)	21.5 (3.7)	25.2 (4.5)	25.2 (4.5)	18.3 (7.2)
Letter	75.4 (0.7)	74.9 (0.8)	80.8 (0.5)	81.0 (0.3)	81.0 (0.4)	82.9 (0.3)	82.9 (0.3)	82.9 (0.3)	74.9 (0.5)
News20	90.9 (0.1)	91.0 (0.2)	91.1 (0.1)	91.0 (0.1)	91.0 (0.1)	90.6 (0.1)	90.6 (0.1)	90.6 (0.1)	89.4 (0.2)
Average	32.1 (0.7)	44.0 (2.3)	49.5 (2.9)	49.6 (2.8)	50.4 (3.0)	48.8 (1.0)	48.2 (2.3)	49.1 (3.6)	47.0 (3.9)

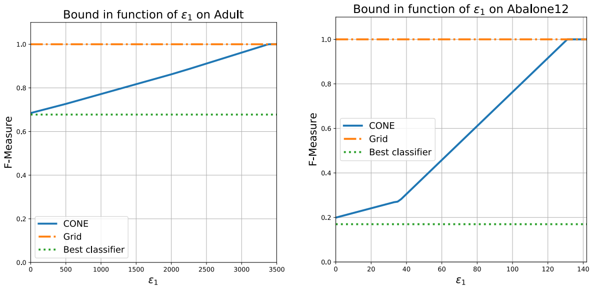

Figure 4: Bounds on the F-measure as a function of ε_1, the unknown sub-optimality of the SVM learning algorithm. Results are shown on two datasets: Adult (left) and Abalone12 (right).

achievable F-measure depending on the value of ε_1, with a hard maximum at 1. The values of ε_1 are expressed in number of points for an easier interpretation.

The bound from Parambath et al. (2014) gives loose guarantees and the aggregate bound is most of the time above 1. The values, before being clipped to 1 can for example start at $F = 7$ and end up at $F = 40$ (on Yeast, if plotted on the same range of ε_1 values). This representation shows once again that our bounds are very tight. On Abalone10 and Letter, where the other bound starts below 1 (see supplementary), the graph also confirms the fact that our bounds aref less sensitive to the value of ε_1 ($\Phi_{\boldsymbol{e}} \leq \Phi$).

Visualizing unreachable zones. The grayed-out areas in Fig. 5 are the unreachable zones. This figure shows that the guarantees obtained with our bounds are much more relevant than the ones from Parambath et al. (2014). Indeed, it is only on two datasets (Abalone10 and Letter, see supplementary) that the previously existing bound actually gives a maximum possible F-measure that is below 1. Our bounds give unreachable zones that go very close to the empirical points.

Looking at the cones with our tight bounds, we see that sometimes a point is in the cone generated by another point. This looks like a violation of our bounds but it rather shows that ε_1 cannot be considered to be 0 in the current setting. Naturally, $\varepsilon_1 \neq 0$ comes from the fact that the weighted-SVM is not robust and not optimal in terms of weighted-0/1 loss. Our intuition is that the SVM is less and less optimal as the weights become more extreme, such as when t gets closer to 0.

Bounds' evolution across iterations. We now study, with **CONE**, how the training performance and the overall bound evolve as we add more models. In **CONE** adding a model means doing one more iteration, while with the grid approach Parambath et al. (2014) it requires to re-learn all models (as all grid locations change). Fig. 6 (and supplementary) illustrates that **CONE** tends to produce better models at a lower cost. These figures also outline the fact that our upper bound is tight and goes down quickly as we add models.

4.3 Performance in F-measure at test time

Finally, we compare the performance of **CONE** (SVM$_C$), based on SVM algorithm against its competitors: LR$_B$ for the method of Narasimhan et al. (2015b), LR$_{I.R.}$ and LR^T for Koyejo et al. (2014) and SVM$_G$/LR$_G$ for the method of Parambath et al. (2014). We present the results of all methods in Tab. 2, giving a budget of 19 models for relevant algorithms. Overall, **CONE** performs at least as well as its competitors

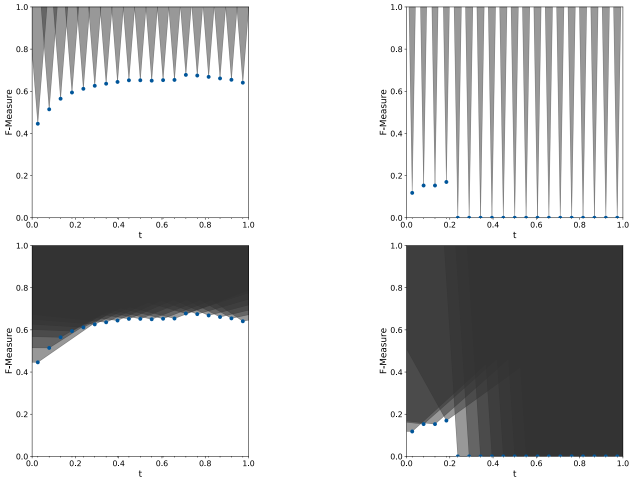

Figure 5: Unreachable region obtained from the same 19 (t, F) points corresponding to learning weighted-SVMs on a grid of t values. Cones are shown for the Adult (left) and Abalone12 (right) datasets, and with the bound from Parambath et al. (2014) (top) and with our tighter bounds presented in Section 2.3 (bottom).

in average, and the very best results are obtained by combining **CONE** with thresholding.

The baseline of using a simple SVM completely fails on half of the datasets. The improved SVM which consists in rebalancing the classes (SVM$_{I.R.}$) still performs worse than other approaches in average, and on most datasets. Even with thresholding, the approaches that learn a single model (LR^T and LR$^T_{I.R.}$) are still outperformed by the ones that learn multiple models with different class-weights like ours (all subscripts $_C$) and the grid one (subscripts $_G$). This last result shows that it is insufficient to solely rely on tuning the threshold of a single model.

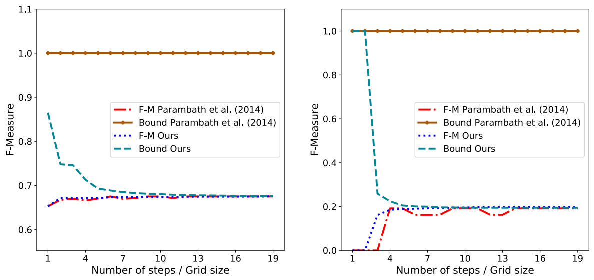

Figure 6: Training performance of **CONE** versus the grid approach from Parambath et al. (2014), together with their respective bounds (on Adult (left) and Abalone12 (right)). We suppose $\varepsilon_1 = 0$, which explains that we observe empirical values that are higher than our upper bound (on Abalone12).

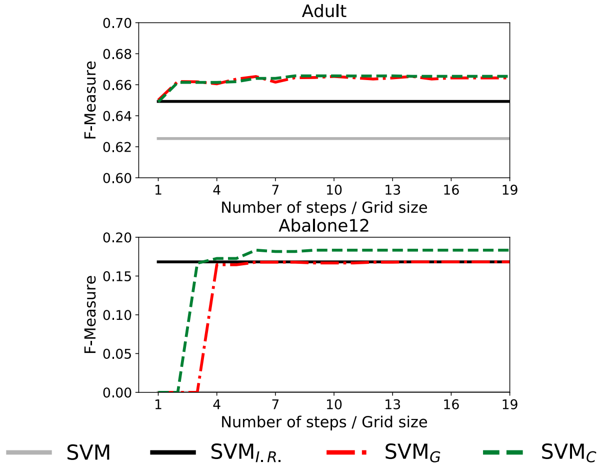

Figure 7: F-measure obtained on the test set for four considered approaches on Adult (top) and Abalone12 (bottom) datasets, plotted as a function of the computing budget (number of weighted SVM to learn).

In average, both our approach and the grid approach outperform all other considered approaches, including the bisection algorithm LR$_B$. We see that the results of **CONE** are very similar to the grid approach SVM$_G$. However, looking at Fig. 7 at the bottom (but also in supplementary), we see that the proposed method is able to reach higher values with a limited number of iterations, i.e. after training fewer models.

5 Conclusion

In this work, we have presented new bounds on the F-measure based on a cost-sensitive classification approach. These bounds have been shown to be tighter than existing ones and less sensitive to the sub-optimality of the learned classifier (ε_1). Furthermore, we have shown that our bounds are useful from a practical point of view by deriving **CONE**, an algorithm which iteratively selects class weights to reduce the overall upper bound on the optimal F-measure. Finally, **CONE** has been shown to perform at least as well as its competitors on various datasets.
If this work focuses on the F-measure, it can be generalized to any other linear-fractional performance measure. Our perspectives include estimating ε_1 (for example using (Bousquet et al., 2004)), refining our framework to improve the search space exploration or to adapt it to SGD-based learning algorithms, and finally deriving generalization guarantees.

Acknowledgements

This work is supported by the TADALoT project (Pack Ambition 2017, 17 011047 01) and the FUI MIVAO.

References

Alberto, C. and Laura, M. (2009). *Generalized Convexity and Optimization: Theory and Applications*. Lecture Notes in Economics and Mathematical Systems 616. Springer-Verlag Berlin Heidelberg, 1 edition.

Bousquet, O., Boucheron, S., and Lugosi, G. (2004). *Introduction to Statistical Learning Theory*, pages 169–207. Springer Berlin Heidelberg, Berlin, Heidelberg.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Busa-Fekete, R., Szörényi, B., Dembczynski, K., and Hüllermeier, E. (2015). Online f-measure optimization. In *NIPS*.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*

Decubber, S., Mortier, T., Dembczynski, K., and Waegeman, W. (2018). Deep f-measure maximization in multi-label classification: A comparative study. page 16.

Dembczyński, K., Kotłowski, W., Koyejo, O., and Natarajan, N. (2017). Consistency analysis for binary classification revisited. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 961–969, International Convention Centre, Sydney, Australia. PMLR.

Dembczynski, K. J., Waegeman, W., Cheng, W., and Hüllermeier, E. (2011). An exact algorithm for f-measure maximization. In *NIPS*.

Jansche, M. (2005). Maximum expected f-measure training of logistic regression models. In *EMNLP*.

Joachims, T. (2005). A support vector method for multivariate performance measures. In *ICML*.

Koyejo, O. O., Natarajan, N., Ravikumar, P. K., and Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2744–2752. Curran Associates, Inc.

Lopez, V., Fernandez, A., Garcia, S., Palade, V., and Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113 – 141.

Musicant, D. R., Kumar, V., Ozgur, A., et al. (2003). Optimizing f-measure with support vector machines. In *FLAIRS*.

Narasimhan, H., Kar, P., and Jain, P. (2015a). Optimizing non-decomposable performance measures: A tale of two classes. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 199–208.

Narasimhan, H., Ramaswamy, H., Saha, A., and Agarwal, S. (2015b). Consistent multiclass algorithms for complex performance measures. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2398–2407, Lille, France. PMLR.

Parambath, S. P., Usunier, N., and Grandvalet, Y. (2014). Optimizing f-measures by cost-sensitive classification. In *NIPS*, pages 2123–2131.

P.M. Chinta, P. Balamurugan, S. S. and Murty, M. (2013). Optimizing f-measure with non-convex loss and sparse linear classifiers. In *IJCNN*.

Rapcsák, T. (1991). On pseudolinear functions. *European Journal of Operational Research*, 50(3):353–360.

van Rijsbergen, C. J. (1974). Further experiments with hierarchic clustering in document retrieval. *Information Storage and Retrieval*, 10(1):1–14.

Ye, N., Chai, K. M. A., Lee, W. S., and Chieu, H. L. (2012). Optimizing f-measure: A tale of two approaches. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*.

Zhao, M.-J., Edakunni, N., Pocock, A., and Brown, G. (2013). Beyond fano's inequality: Bounds on the optimal f-score, ber, and cost-sensitive risk and their implications. *JMLR*.