CONE : A Cost-Sensitive Classification Wrapper for Iterative F-Measure Optimization

K. Bascol[1,3], R.Emonet[1], E. Fromont[2], A.Habrard [1], G. Metzler[1,4], and M. Sebban[1]

1. Univ. Lyon, UJM-Saint-Etienne, Laboratoire Hubert Curien UMR CNRS 5516, F-42023, Saint-Etienne, France
2. Univ. Rennes 1, IRISA/Inria, 35042 Rennes cedex, France
3. BLUECIME inc., France and 4. BLITZ inc., France

Abstract

In an **imbalanced** setting:
$\longrightarrow$ optimizing the classical accuracy tends to predict only the majority class;
$\longrightarrow$ optimizing imbalance-proof measures (as the F-Measure) is a tough task due to its non-convexity;
$\Longrightarrow$ **Solution**: approximate F-measure optimization by cost-sensitive approach.

Based on S. P. Parambath et al. work [1] and driven by theoretical guarantees, to tackle imbalanced problems we propose:

- a tighter bound than the one given in [1];
- $\mathbf{CONE}$, an algorithm in which the weights of each classes are updated iteratively;
- a way to prune the search space of the weights for low values of F-Measure.

Notations and Base Result

Context and Notations

- Only *binary* setting is presented, but our bounds can be derived to be used in a *multi-class* setting.
- $\boldsymbol{e} = (e_1, e_2) = (FN, FP)$ the error profile,
- $F = \dfrac{(1+\beta^2)(P-e_1)}{(1+\beta^2)P - e_1 + e_2}$ the F-Measure,
- $a(t) = (1+\beta^2-t, t)$ a weighting function, assigns cost of missclassification on each classes,
- g an evaluation of a for a value of t,
- $h \in \mathcal{H}$ a classifier,
- ε_0 upper bound on the norm between two evaluations of a,
- ε_1 the sub-optimality of a classifier,
- Φ a kind of Lipschitz constant on F.

Base Result [1]

Let $\varepsilon_0 \geq 0$ and $\varepsilon_1 \geq 0$, and assume that there exists $\Phi > 0$ such that for all $\boldsymbol{e}, \boldsymbol{e}'$ satisfying $F(\boldsymbol{e}') > F(\boldsymbol{e})$, we have:

$$F(\boldsymbol{e}') - F(\boldsymbol{e}) \leq \Phi \langle a(F(\boldsymbol{e}')), \boldsymbol{e} - \boldsymbol{e}' \rangle.$$

Then , let us take $\boldsymbol{e}^\star \in argmax\ F(\boldsymbol{e}')$ and denote $\boldsymbol{a}^\star = a(F(\boldsymbol{e}^\star))$. Let furthermore $\boldsymbol{g} \in \mathbb{R}^d_+$ and $h \in \mathcal{H}$ satisfying the following two conditions:

$$(i)\ \|\boldsymbol{g} - \boldsymbol{a}^\star\|_2 \leq \varepsilon_0, \quad (ii)\ \langle \boldsymbol{g}, \mathbf{E}(h) \rangle \leq \min_{\boldsymbol{e}' \in \mathcal{E}(\mathcal{H})} \langle \boldsymbol{g}, \boldsymbol{e}' \rangle + \varepsilon_1.$$

We have: $F(\mathbf{E}(h)) \geq F(\boldsymbol{e}^\star) - \Phi(2\varepsilon_0 M + \varepsilon_1), \quad M = \max_{\boldsymbol{e}' \in \mathcal{E}(\mathcal{H})} \|\boldsymbol{e}'\|_2,$

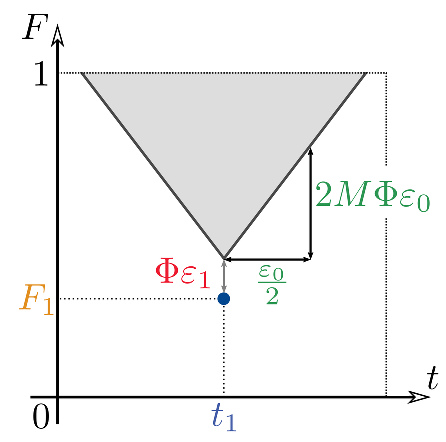
where $F(\boldsymbol{e}^\star)$ is the optimal value of the F-Measure.

Geometric Interpretation

According to [1], $\|a(t_1) - a(t)\|_2 \leq 2\|t_1 - t\|_2 = \varepsilon_0$, the bound can be rewritten as follows:

$$F(\boldsymbol{e}(t)) \leq F(\boldsymbol{e}(t_1)) + 4\Phi M\|t_1 - t\|_2 + \Phi\varepsilon_1.$$

This can be identified as the definition of Lipschitz function applied to the F-Measure with respect to t, with a Lipschitz constant equal to $4\Phi M$ and an offset of $\Phi\varepsilon_1$.



CONE: a bound driven search algorithm

Let us consider $t, t_1 \in [0, 1]$ be two values used to assign costs and $\boldsymbol{e}(t), \boldsymbol{e}(t_1)$ the vector of miss-classified examples. Under the assumptions of the *Base Result* and using the same notations we have :

A tighter slope:

$$F(\boldsymbol{e}(t)) \leq F(\boldsymbol{e}(t_1)) + \Phi(\sqrt{2}(\|\boldsymbol{e}(t_1)\|_2 + M')\|t_1 - t\|_2) + \Phi\varepsilon_1.$$

In other words, we refined the slope of the cones to $\sqrt{2}\Phi\left(\|\boldsymbol{e}(t_1)\|_2 + M'\right)$, where M' is defined as:

$$\max_{\boldsymbol{e}'} \|\boldsymbol{e}'\|_2 \quad s.t \quad F(\boldsymbol{e}') > F(\boldsymbol{e}(t_1)).$$

Furthermore, if $t > t_1$ then :

Search space pruning:

$$F_\beta(\boldsymbol{e}(t)) \leq (1+\beta^2)\frac{\frac{1+\beta^2}{t_1}TP(t_1)}{\beta^2\frac{1+\beta^2}{t_1}TP(t_1) + P}.$$

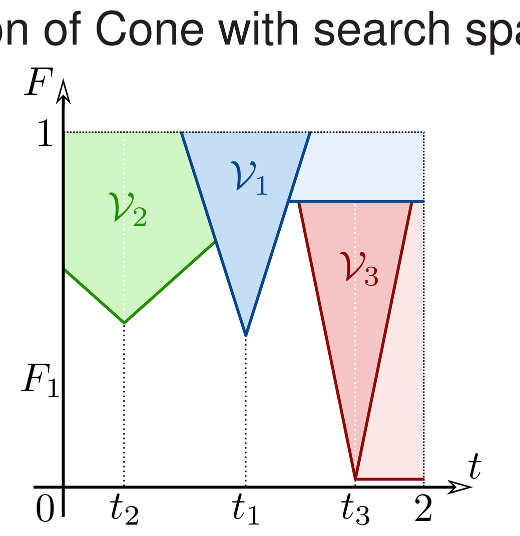
Intuition: if TP small, decreasing the weights on the Positive class shouldn't be beneficial.

CONE Algorithm

Input: β, //F-measure parameter
Input: S, //training set
Input: $wLearn$, //weighted-learning algorithm
Input: $shouldStop$. //stopping criterion

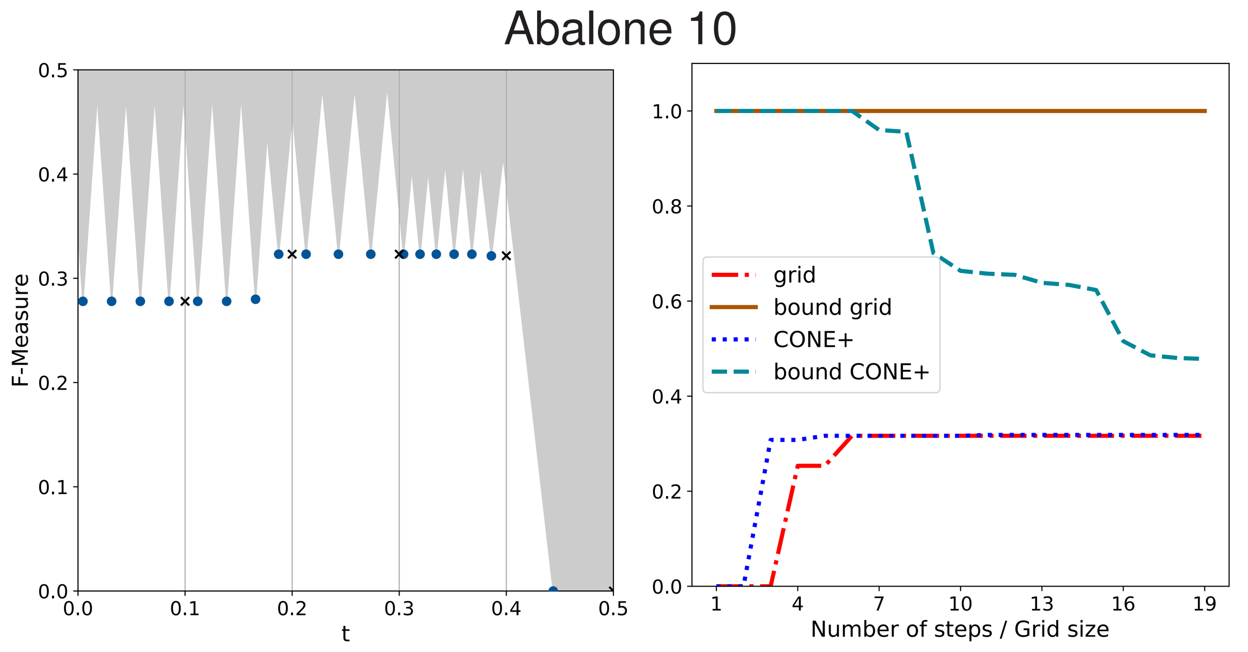
Initialize $i = 0$ //iteration number
Initialize $\mathcal{Z}_0 = \varnothing$ //excluded zones
repeat
 $i = i + 1$
 $t_i = findNextT(\mathcal{Z}_{i-1})$
 $classifier_i = wLearn(1 + \beta^2 - t_i, t_i)$
 $F_i = F_\beta(classifier_i, S)$
 $\mathcal{V}_i = unreachableZone(t_i, F_i, S)$
 $\mathcal{Z}_i = \mathcal{Z}_{i-1} \cup \mathcal{V}_i$
until $shouldStop(i, classifier_i, \mathcal{Z}_i)$

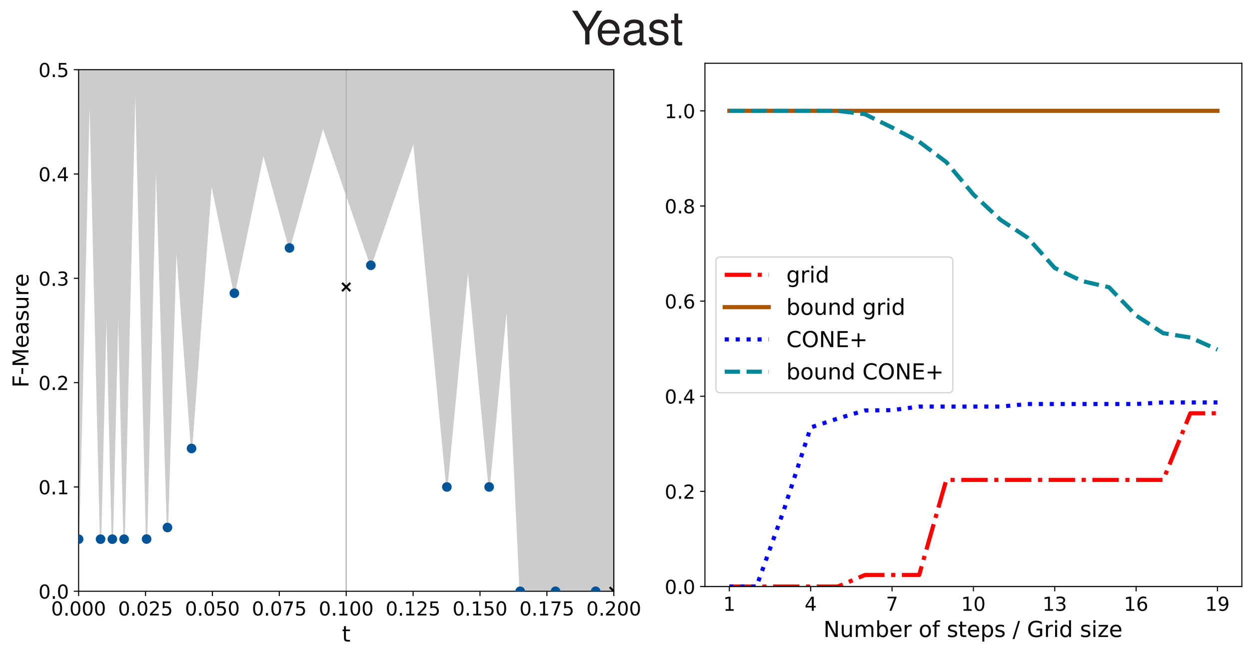
An illustration of Cone with search space pruning



ν_1: First cone halves the search space: $t_1 = 1$
$\rightarrow$ Highest remaining $F = 1$ for $t \in [0,\ 0.6]$
ν_2: Next cone halves this interval: $t_2 = 0.3$
$\rightarrow$ Highest remaining $F = 0.7$ for $t \in [1.3,\ 2]$
ν_3: Next cone halves this interval: $t_3 = 1.65$
$\rightarrow$ Highest remaining $F = 0.7$ for $t \in [1.3,\ 1.35]$
ν_∞: Until we reach the best F possible

Practical Evaluation of Theoretical Guarantees





Examples of runs of our method (blue points and shaded area) and of the **grid** wrapper (black crosses) both with a cost-sensitive SVM classifier with $C = 1$. We also represent the corresponding evolution of the F-Measure and of the considered bound as a function of the number of **CONE** steps and [1] grid size.

F_1-Measure for Logistic Regression and SVM algorithms (averaged over 5 experiments). Number of CONE steps and grid size are both limited to 9 and 4 SVMs.

Steps	9			4		
Dataset	SVM*	SVM$_C$	SVM$_C^+$	SVM*	SVM$_C$	SVM$_C^+$
Adult	66.1 (0.1)	66.5 (0.1)	66.5 (0.1)	65.8 (0.3)	66.5 (0.03)	66.5 (0.04)
Abalone10	30.2 (2.5)	31.0 (1.1)	32.3 (1.2)	30.7 (2.8)	12.2 (14.5)	30.8 (1.1)
IJCNN'01	61.6 (0.4)	61.0 (0.6)	61.6 (0.6)	61.0 (0.5)	61.0 (0.6)	61.0 (0.6)
Abalone12	16.1 (3.5)	12.2 (7.0)	17.0 (3.5)	0.0 (0.0)	0.0 (0.0)	15.9 (3.7)
Yeast	24.5 (16.3)	34.8 (8.3)	32.3 (12.2)	33.0 (18.0)	14.7 (12.0)	35.0 (8.4)
Wine	11.7 (11.3)	11.3 (10.8)	19.4 (6.6)	0.0 (0.0)	0.0 (0.0)	17.7 (4.4)

"$\star$": Reproduction of [1] ; "$_C$": $\mathbf{CONE}$; "^+_C": $\mathbf{CONE}$ with pruning method

Conclusion

In this work, we derive a tighter bound than the one obtained [1]. Moreover, combining it with a search space algorithm we manage to match, and even outmatch, [1] method with less classifiers and without needing an arbitrary sized grid search.
We now aim to derive a similar search space pruning on the left (i.e. for smaller values of t). We also aim to extend the applications, using neural networks for instance and see how to deal with the notion of sub-optimality in the non convex cases.

Reference

[1] . P. Parambath, N. Usunier and Y. Grandvalet, Optimizing f-measures by cost-sensitive classification, *NIPS* 2014.

Acknowledgements

LABORATOIRE HUBERT CURIEN
RENNES
bluecime