Optimizing the Multiclass F-measure via Biconcave Programming

Weiwei Pan, Harikrishna Narasimhan
Harvard University, USA
{*weiweipan, hnarasimhan*}*@seas.harvard.edu*

Purushottam Kar
IIT Kanpur, India
purushot@cse.iitk.ac.in

Pavlos Protopapas
Harvard University, USA
pavlos@seas.harvard.edu

Harish G. Ramaswamy
IBM Research, India
harishguruprasad@gmail.com

***Abstract*—The F-measure and its variants are performance measures of choice for evaluating classification and retrieval tasks in the presence of severe class imbalance. It is thus highly desirable to be able to directly optimize these performance measures on large-scale data. Recent advances have shown that this is possible in the simple binary classification setting. However, scant progress exists in multiclass settings with a large number of classes where, in addition, class-imbalance is much more severe. The lack of progress is especially conspicuous for the *macro-averaged* F-measure, which is the widely preferred F-measure variant in multiclass settings due to its equal emphasis on rare classes. Known methods of optimization scale poorly for macro F-measure, often requiring run times that are exponential in the number of classes.**

We develop BEAM-F, the first efficient method for directly optimizing the macro F-measure in multiclass settings. The challenge here is the intractability of optimizing a sum of fractional-linear functions over the space of confusion matrices. We overcome this difficulty by formulating the problem as a biconcave maximization program and solve it using an efficient alternating maximization approach that involves a Frank-Wolfe based iterative solver. Our approach offers guaranteed convergence to a stationary point and experiments show that, for a range synthetic data sets and real-world applications, our method offers superior performance on problems exhibiting large class imbalance.

***Keywords*-F-measure; Class imbalance; Multiclass classification; Alternating maximization; Frank-Wolfe method**

I. Introduction

In real world classification tasks, ranging from medical diagnosis to web-scale document tagging, one frequently encounters cases of severe imbalance in the classes. The thrust in all these applications is on correctly labeling the rare classes. Performance measures such as classification accuracy do not satisfy this goal, as they unreasonably reward a naive classifier that classifies all points to a majority class. Performance measures that do account for rare classes are more involved, and evaluate classifiers in a more holistic manner. An excellent example of such measures is the *F-measure*, the harmonic mean of the *precision* and *recall* of a classifier [1]. The F-measure has enjoyed prominence in learning and retrieval literature and several variants exist. For multiclass settings, a popular variant is to compute the F-measure for the binary one-vs-all problem for each class and then take their average. This version, known as the *macro-averaged* F-measure, is very popular in many classification and retrieval problems [2] due to its emphasis on performing well on rare classes.

However, the macro F-measure is extremely challenging to optimize directly, owing to its complicated *sum of fractional linear* structure. Existing approaches often attempt to extend techniques used in optimizing F-measure in the *binary classification* setting to the multiclass setting. Of these techniques, there exist three broad classes: (a) plug-in methods [3]–[6] which learn a *class probability estimation* model and then tune a threshold over the estimates to obtain a classifier, (b) the structural SVM technique that optimizes a concave surrogate to the F-measure [7], and (c) stochastic optimization techniques that work with a pseudo-concave surrogate for the F-measure [8], [9].

However, multiclass extensions of all these techniques fail. The plug-in method for macro F-measure optimization fails as it requires a joint tuning of $\Omega(n^2)$ thresholds, where n is the number of classes, which takes time exponential in n. The structural SVM method also takes exponential time for subgradient computations. The stochastic methods do not apply to multiclass settings since the pseudo-concave structure they assume is missing in the macro F-measure.

There has been some work on optimizing the *micro-averaged* F-measure in multiclass settings [10], as well as optimizing the macro F-measure in *multilabel classification* settings [11]–[14]. However, these do not extend to multiclass settings. The work of [10] assumes a pseudo-concave structure that macro F-measure does not have. The multilabel work on the other hand is simply inapplicable. In multilabel learning, labels can coexist, while in multiclass settings, classes are exclusive. This non-exclusivity decomposes the problem and allows a simple one-vs-all approach to optimize the multilabel macro F-measure. The multiclass macro F-measure is in fact a *sum of fractional-linear functions* of the confusion matrix – neither pseudo concave nor decomposable. In fact, the problem of optimizing a sum of fractional-

linear functions is known to be NP-hard in general [15].

In this paper, we develop BEAM-F, the first efficient learning algorithm for directly optimizing the multiclass macro F-measure. Our main contribution is that of building upon the framework of [10] and adapting intuition from optimization literature [16] to cast the macro F-measure optimization problem as a *biconcave program*. This allows us to utilitze tools for optimizing concave performance measures [10] to solve the macro F-measure optimization problem, using an alternating maximization-based solver. We show that BEAM-F provably converges to the stationary points of the optimization problem. In our experiments, we found plug-in and structural SVM approaches too expensive to benchmark, something that has been observed before [10]. Nevertheless, on a range of real life applications and synthetic data sets, BEAM-F demonstrates superior performance compared to existing standard approaches to multiclass classification, especially when the class imbalance is excessive and the class overlap is significant.

II. Problem Setting and Preliminaries

Consider an instance space $\mathcal{X}$ and label space $[n] = \{1, \ldots, n\}$. We wish to learn a classifier $h : \mathcal{X} \to [n]$ that maps instances in $\mathcal{X}$ to one of the labels. We assume that an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times [n]$ generates the data. Let $\eta_i(x) = \mathbf{P}(y = i \,|\, x)$ denote the conditional class probability, and $p_i = \mathbf{P}(y = 1)$ be the class proportions. We will be interested in settings where the class proportions p_i deviate significantly from $1/n$, resulting in class imbalance.

Classification accuracy. A standard approach for evaluating a classifier is to evaluate its classification accuracy, the fraction of correctly labeled instances $\text{Acc}[h] = \mathbf{P}_{(x,y)\sim\mathcal{D}}(h(x) = y)$. This evaluation measure is not well-suited for class imbalanced settings, as a classifier that performs well only on popular classes will still yield high classification accuracy. For such class imbalanced settings, measures that give equal attention to all classes are required.

Precision and Recall. For a classifier h, its recall with respect to class i is the fraction of instances of that class that are correctly classified i.e. $\text{Recall}_i[h] = \mathbf{P}(h(x) = i \,|\, y = i)$. The precision with respect to class i is the fraction of points classified by h as belonging to class i that in fact do belong to i, i.e. $\text{Prec}_i[h] = \mathbf{P}(y = i \,|\, h(x) = i)$. Note that a naive classifier that predicts i on all instances will have a recall of 1, but will have low precision; similarly, a classifier that never predicts i will have a precision of 1, but will have zero recall.

Binary F-measure. In the binary setting ($n = 2$), the harmonic mean of the precision and the recall of a classifier $F_1[h] = \frac{2\times\text{Prec}_1[h]\times\text{Recall}_1[h]}{\text{Prec}_1[h]+\text{Recall}_1[h]}$ define its F-measure performance. The F-measure takes a value between 0 and 1, and higher values imply better performance.

Multiclass F-measure Variants. The *micro F-measure* extends F-measure to multiclass settings by using precision and recall values averaged across all classes [2], or by treating one class as negative and others as positive, and computing the binary F-measure [5]. However, both these variants do not place enough emphasis on rare classes, and are thus not of much use in class imbalanced settings. The more useful variant is the *macro F-measure*, which instead computes the average F-measure across all one-vs-all problems as follows:

$$F_1^{\text{macro}}[h] = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times \text{Prec}_i[h] \times \text{Recall}_i[h]}{\text{Prec}_i[h] + \text{Recall}_i[h]}.$$

The macro F-measure is popular in class imbalanced settings and will be the focus of this paper. Given a training sample $S = \{(x_1, y_1), \ldots, (x_N, y_N))\} \sim \mathcal{D}$, our goal would be to learn a classifier $\widehat{h}$ with maximum macro F-measure.

Confusion matrix. For a classifier h, its confusion matrix $\mathbf{C}[h] \in [0,1]^{n\times n}$ is defined as $C_{ij}[h] = \mathbf{P}(y = i,\, h(x) = j)$. Given a sample S, the *empirical confusion matrix* $\widehat{\mathbf{C}}[h; S]$ is defined as $\widehat{C}_{ij}[h; S] = \frac{1}{N}\sum_{k=1}^{N}\mathbf{1}(y_k = i, h(x_k) = j)$. The empirical confusion matrix acts as a proxy for the true confusion matrix while executing plug-in approaches. We will rewrite our performance measures in terms of the confusion matrix. The binary F-measure can be written as a function of the 2×2 confusion matrix as $F_1[h] = \frac{2C_{11}[h]}{2C_{11}[h]+C_{10}[h]+C_{01}[h]}$. Thus, F-measure has a *fractional-linear* form. It is the ratio of two linear functions of the confusion matrix. One can similarly write the macro F-measure as

$$F_1^{\text{macro}}[h] = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times C_{ii}[h]}{\sum_{j=1}^{n} C_{ij}[h] \,+\, \sum_{j=1}^{n} C_{ji}[h]},$$

The macro F-measure is a sum of fractional-linear functions over the confusion matrix and is non-convex in general.

We now give an overview of plug-in approaches for binary and multiclass classification.

Plug-in for Binary F-measure. The plug-in approach for maximizing F-measure in binary classification involves first learning a class probability model $\widehat{\eta}_1 : \mathcal{X} \to [0,1]$ that estimates the conditional probability $\widehat{\eta}_1(x) \approx \mathbf{P}(y = 1|x)$ of instances, and then constructing a classifier $\widehat{h}(x) = \text{sign}(\widehat{\eta}_1(x) - \theta)$, by tuning a threshold $\theta \in [0,1]$ that yields the maximum F-measure on a validation set. The method has recently received much attention and is known to converge to the optimal classifier for the binary F-measure [4], [6].

Plug-in for Multiclass linear measures. A similar two-step approach can also be applied in multiclass settings for *linear* performance measures of the form $\psi^{\text{linear}}(\mathbf{C}) = \sum_{i,j=1}^{n} W_{ij}C_{ij}$. The weights $W_{ij} \in \mathbb{R}_+$ encode the reward or penalty for classifying an instance of class i as class j. In this case, one can learn a multiclass class probability model $\widehat{\eta} : \mathcal{X} \to \Delta_n$ which estimates the probability of an instance x belonging to class i: $\widehat{\eta}_i(x) \approx \mathbf{P}(y = i|x)$, and then construct a classifier as $\widehat{h}(x) \in \text{argmax}_{y\in[n]}\sum_{i=1}^{n}\eta_i(x)W_{iy}$. This classifier enjoys consistency properties as well [10].

It is tempting to apply this approach for optimizing non-linear performance measures, such as micro and macro F-measure, but such an approach is intractable. The threshold tuning step in this case requires simultaneously tuning $\Omega(n^2)$ thresholds on a grid over $[0,1]^{n^2}$, a task that is exponential in n^2 and hence computationally infeasible. [10] were able to overcome this problem for the micro F-measure by exploiting its pseudo-concave structure. However, no such structure exists for the macro F-measure.

III. Biconcave Optimization for F-measure

A critical step in our BEAM-F algorithm is developing a workaround to the threshold tuning step for the macro F-measure. We do this by extending the plug-in based framework introduced by [10] for optimizing multiclass performance measures. In framework, the learning problem is cast as optimization problem over a space of feasible confusion matrices. However, we reiterate that the work of [10] does not directly apply to the macro F-measure, which is not pseudo-concave. In the following, we develop this idea in steps. Due to space constraints, we omit the proofs here.

Feasible confusion matrices. For a given distribution $\mathcal{D}$, let $\mathcal{C}_\mathcal{D} = \{\mathbf{C}[h] \,|\, h : \mathcal{X} \to [n]\}$ be the set of all confusion matrices that correspond to some classifier. There may exist confusion matrices that are infeasible with respect to $\mathcal{D}$. For example, unless there exists a perfect classifier for $\mathcal{D}$, the identity matrix will not be in its feasible set. The problem of maximizing the macro F-measure over $\mathcal{D}$ can be cast as an equivalent optimization problem over the set of feasible confusion matrices: $\max_{\mathbf{C}\in\mathcal{C}_\mathcal{D}} \psi^{\text{macro}}(\mathbf{C})$, where $\psi^{\text{macro}}(\mathbf{C}) = \frac{1}{n}\sum_{i=1}^n \frac{2\times C_{ii}}{\sum_{j=1}^n C_{ij}+\sum_{j=1}^n C_{ji}}$. We optimize ψ^{macro} using the training sample S as a proxy for D. To ensure that the constraint set is convex, we relax the problem by replacing $\mathcal{C}_\mathcal{D}$ with its convex hull $\overline{\mathcal{C}}_\mathcal{D}$:

$$\max_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D}} \psi^{\text{macro}}(\mathbf{C}). \qquad \text{(OP1)}$$

Since $\overline{\mathcal{C}}_\mathcal{D}$ is the set of all confusion matrices generated by *randomized classifiers* [10], (OP1) seeks to learn a randomized classifier that maximizes the macro F-measure. Recall that the plug-in technique cannot be used to solve this optimization problem, as the macro F-measure is non-linear. Moreover, standard techniques such as projected gradient descent are inapplicable, since the constraint set $\overline{\mathcal{C}}_\mathcal{D}$ is not available explicitly. While *oracle* and *interior-point* methods do not require explicit access to the constraint set, these method apply only to concave and pseudo-concave optimization [10]. The main contribution of this paper, is a biconcave optimization approach for maximizing the macro F-measure.

Fractional-linear Functions. As a warmup, consider optimizing a single fractional-linear function of the confusion matrix $\max_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D}} \frac{\langle \mathbf{A},\mathbf{C}\rangle}{\langle \mathbf{B},\mathbf{C}\rangle}$. Observe that for any $\alpha > 0$, $\max_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D}} \frac{\langle \mathbf{A},\mathbf{C}\rangle}{\langle \mathbf{B},\mathbf{C}\rangle} \geqslant \alpha$ iff $\max_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D}} \langle \mathbf{A},\mathbf{C}\rangle - \alpha\langle \mathbf{B},\mathbf{C}\rangle \geqslant 0$. Thus, to verify that maximum achievable performance level exceeds α, we need only solve the linear maximization problem $\max_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D}} \langle \mathbf{A} - \alpha\mathbf{B}, \mathbf{C}\rangle$, and check if the optimal value is non-negative. To find the optimal performance value and the corresponding classifier, one can then simply do a binary search over values of $\alpha \in [0,1]$.

Sum of Fraction-linear Functions. Doing the above with a sum of more than one fractional linear function is not feasible since we would have to tune α_i for each function jointly, which is intractable [15]. We adopt a totally different strategy, formulating (OP1) as an equivalent biconcave optimization problem over certain auxiliary variables and the confusion matrix [16]. It is easy to see that the macro F-measure can be rewritten as $\psi^{\text{macro}}(\mathbf{C}) = \frac{1}{n}\sum_{i=1}^n \frac{\langle \mathbf{A}^i,\mathbf{C}\rangle}{\langle \mathbf{B}^i,\mathbf{C}\rangle}$, where, for each class i, the matrix A^i satisfies $A^i_{jk} = 1$ if $j = k = i$ and 0 otherwise, and where B^i satisfies $B^i_{ii} = 2$, $B^i_{ij} = B^i_{ji} = 1$ and $B^i_{jk} = 0$ for all $j,k \neq i$. We introduce an auxiliary variable $u_i \in \mathbb{R}_+$ for each class i. These variables are analogous to the level variable α used previously in optimizing a single fractional-linear function. Consider the following biconcave function $\psi^{\text{biconcave}}(\mathbf{u},\mathbf{C}) = \frac{1}{n}\sum_{i=1}^n \left[2u_i\sqrt{\langle \mathbf{A}^i,\mathbf{C}\rangle} - u_i^2\langle \mathbf{B}^i,\mathbf{C}\rangle\right]$. One can verify that for a fixed value of $\mathbf{u} \in \mathbb{R}^n_+$, the function is concave with respect to $\mathbf{C}$, and that for a fixed $\mathbf{C} \in \overline{\mathcal{C}}_\mathcal{D}$, the function is concave in $\mathbf{u}$. Our original optimization problem (OP1) can then be shown to be equivalent to the following

$$\max_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D},\, \mathbf{u}\in\mathbb{R}^n_+} \psi^{\text{biconcave}}(\mathbf{u},\mathbf{C}). \qquad \text{(OP2)}$$

Theorem 1. *The optimization problems* OP1 *and* OP2 *have the same set of maximizers over* $\overline{\mathcal{C}}_\mathcal{D}$.

This theorem is a special case of a result in [16].

IV. Alternating Maximization Solver

We now develop the BEAM-F method to solve the biconcave program in (OP2). BEAM-F adopts an alternating maximization approach, alternately fixing the auxiliary variable and the confusion matrix, and updating the other. BEAM-F starts with an initial $\mathbf{C}^0$ and loops over Steps I, II.

Step I. $\mathbf{u}^t \in \text{argmax}_{\mathbf{u}\in\mathbb{R}^n_+} \psi^{\text{biconcave}}(\mathbf{u},\mathbf{C}^{t-1})$. This step admits a closed form solution $u^t_i = \frac{\sqrt{\langle \mathbf{A}^i,\mathbf{C}^{t-1}\rangle}}{\langle \mathbf{B}^i,\mathbf{C}^{t-1}\rangle}$.

Step II: $\mathbf{C}^t \in \text{argmax}_{\mathbf{C}\in\overline{\mathcal{C}}_\mathcal{D}} \psi^{\text{biconcave}}(\mathbf{u}^t,\mathbf{C})$. We use a Frank-Wolfe method [10] to solve this concave problem.

Frank-Wolfe Based Concave Maximization The Frank-Wolfe (FW) algorithm is a provably convergent technique for solving concave maximization problems over a convex set [17], when the constraint set is not explicitly available but linear maximization over the set is nevertheless possible. This fits very well with our setting: although $\overline{\mathcal{C}}_\mathcal{D}$ is not accessible, as we show below, linear maximization over this set can be performed efficiently using the plug-in approach. BEAM-F exploits this fact to maximize the concave objective $\psi^{\text{biconcave}}(\mathbf{u}^t,\mathbf{C})$ over $\overline{\mathcal{C}}_\mathcal{D}$, by formulating a sequence of

linear approximations to the objective. More specifically, let $\mathbf{u}^t$ be the current value of the auxiliary variables. The FW algorithm starts with an initial confusion matrix $\mathbf{\Gamma}^{(t,0)}$ and at each sub-iteration (t,s), computes the gradient of the concave objective at $\mathbf{\Gamma}^{(t,s)}$: $\mathbf{G}^{(t,s)} = \nabla_{\mathbf{C}} \psi^{\text{biconcave}}(\mathbf{u}^t, \mathbf{\Gamma}^{(t,s)})$, and maximizes the corresponding linear approximation of the objective over $\overline{\mathcal{C}}_{\mathcal{D}}$: $\max_{\mathbf{C} \in \overline{\mathcal{C}}_{\mathcal{D}}} \langle \mathbf{G}^{(t,s)}, \mathbf{C} \rangle$. Now although it is not clear how to optimize over $\overline{\mathcal{C}}_{\mathcal{D}}$, the following result shows that it is enough to optimize $\langle \mathbf{G}, \mathbf{C} \rangle$ only over $\mathbf{C} \in \mathcal{C}_{\mathcal{D}}$, which can be done using the plug-in approach.

Theorem 2. *For any* $\mathbf{G}$, $\operatorname{argmax}_{\mathbf{C} \in \mathcal{C}_{\mathcal{D}}} \langle \mathbf{G}, \mathbf{C} \rangle \subseteq \operatorname{argmax}_{\mathbf{C} \in \overline{\mathcal{C}}_{\mathcal{D}}} \langle \mathbf{G}, \mathbf{C} \rangle$.

The plug-in approach (1) divides the training set into two parts, $S = (S_1, S_2)$, (2) uses S_1 to learn a class probability model $\widehat{\eta} : \mathcal{X} \to \Delta_n$, and (3) at any sub-iteration (t,s), constructs the classifier $h^{(t,s)}(x) \in \operatorname{argmax}_{y \in [n]} \sum_{i=1}^n \widehat{\eta}_i(x) G_{iy}^{(t,s)}$ and estimates a new confusion matrix $\widehat{\mathbf{C}}[h^{(t,s)}; S_2]$.

The next iterate $\mathbf{\Gamma}^{(t,s+1)}$ of the FW solver is then set to a suitable convex combination of the previous iterate, and the new confusion matrix determined through a line search. An outline of the BEAM-F solver is provided in Algorithm 1.

BEAM-F finally outputs a randomized classifier by combining the deterministic classifiers obtained across different iterations, along with their corresponding weights.

Running time. Whereas the plug-in and structural SVM approaches require exponential run-time, BEAM-F only requires $O(n^2)$ time per iteration. In practice, BEAM-F was always found to converge in a small number of iterations.

Convergence Guarantees Existing convergence analyses for alternating minimization [18], [19] do not apply directly to BEAM-F since those analyses require exact updates in the alternations, which the Frank-Wolfe solver is unable to offer due to its inability to access the data distribution $\mathcal{D}$. Nevertheless, we can show that the iterates generated by BEAM-F converge to an approximate stationary point $(\widehat{\mathbf{u}}, \widehat{\mathbf{C}})$ such that $\|\nabla \psi^{\text{biconcave}}(\widehat{\mathbf{u}}, \widehat{\mathbf{C}})\|_2 \leqslant \epsilon$, where ϵ diminishes to 0 as the number of training points $N \to \infty$. We omit the details of this result for lack of space.

Algorithm 1 BEAM-F: BiconcavE progrAmming for Macro F-measure optimization

Input: $S = ((x_1, y_1), \ldots, (x_N, y_N))$
Parameter: $T_{\text{out}}, T_{\text{in}}$
Split S into S_1 and S_2
Learn class probability model $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ using S_1
Initialize: $h^{0,0} : \mathcal{X} \to [n]$, $\mathbf{C}^0 = \widehat{\mathbf{C}}[h^{0,0}; S_2]$
For $t = 1$ **to** T_{out} **do**
  Step I: Maximize over $\mathbf{u}$ for fixed $\mathbf{C}^{t-1}$
    $u_i^t = \dfrac{\sqrt{\langle \mathbf{A}^i, \mathbf{C}^{t-1} \rangle}}{\langle \mathbf{B}^i, \mathbf{C}^{t-1} \rangle}$
  Step II: Maximize over $\mathbf{C}$ for fixed $\mathbf{u}^t$
    $\mathbf{\Gamma}^{(t,0)} = \mathbf{C}^t$
    For $s = 1$ **to** T_{in} **do**
      $\mathbf{G}^{(t,s)} = \nabla_{\mathbf{C}} \psi^{\text{biconcave}}(\mathbf{u}^t, \mathbf{\Gamma}^{(t,s-1)})$
      Construct $h^{(t,s)}(x) \in \operatorname{argmax}_{y \in [n]} \sum_{i=1}^n \widehat{\eta}_i(x) G_{iy}^{(t,s)}$
      $\mathbf{C}' = \widehat{\mathbf{C}}[h^{(t,s)}; S_2]$
      $\alpha^{(t,s)} \in \operatorname{argmax}_{\alpha \in [0,1]} \psi^{\text{biconcave}}\big((1-\alpha)\mathbf{\Gamma}^{(t,s-1)} + \alpha\, \mathbf{C}'\big)$
      $\alpha^{(t,\ell)} = \alpha^{(t,\ell)} \alpha^{(t,s)}\ \forall \ell = 1, \ldots, s-1$
      $\mathbf{\Gamma}^{(t,s)} = (1 - \alpha^{(t,s)}) \mathbf{\Gamma}^{(t,s-1)} + \alpha^{(t,s)}\, \mathbf{C}'$
    End For
    $\mathbf{C}^{t+1} = \mathbf{\Gamma}^{(t,T_{\text{in}})}$
End For
Output: A randomized classifier constructed from $h^{(1,1)}, \ldots, h^{(T_{\text{out}}, T_{\text{in}})}$ that for any instance $x \in \mathcal{X}$ predicts $h^{(t,s)}(x)$ with probability $\alpha^{(t,s)}$

V. Experiments

We empirically evaluated the BEAM-F method on a range of synthetic and benchmark data sets, and a case study on celestial object classification[1].

BEAM-F. The class probability model in BEAM-F was learned using a multiclass (softmax) logistic regression technique. Since BEAM-F seeks to optimize a non-convex objective, we use multiple restarts and choose the classifier resulting in the best performance on the training set. BEAM-F was initialized using confusion matrices from plain logistic regression, balanced logistic regression model, and three random classifiers. The inner and outer loops of BEAM-F were terminated upon observing insufficient improvement in the objective. In practice, we found BEAM-F to converge within 3–7 inner and outer iterations. For stability, we used a regularized biconcave function $\widetilde{\psi}(\mathbf{u}, \mathbf{C}) = \psi^{\text{biconcave}}(\mathbf{u}, \mathbf{C}) - 0.5 \cdot \lambda \|\mathbf{C}\|_F^2$.

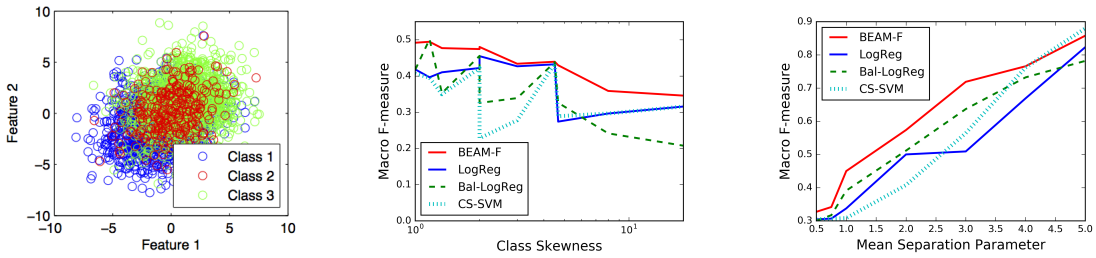
Baselines. We found the plug-in and structural SVM approaches too expensive to benchmark [10] and hence compared against three standard multiclass methods:

1) *LogReg:* A standard multiclass logistic regression model with equal weights on all classes.
2) *Bal-LogReg:* A balanced version of the multiclass logistic regression model where the classifier is constructed by performing a weighted argmax on the estimated probabilities, with the weight for class i set to the the inverse of its proportion in the train set p_i with the aim to place larger emphasis on rare classes.
3) *CS-SVM:* A standard multiclass SVM based on the Crammer-Singer surrogate from the LIBLINEAR library[2].

In each case, we used 70% of the data for training, and the remaining for testing. The parameters for all baselines

[1]Code available at: https://github.com/onefishy/BEAM-F

[2]https://www.csie.ntu.edu.tw/~cjlin/liblinear/



(a) (b) (c)

Figure 1. Experiments on synthetic data: (a) sample data depicting three overlapping classes; a comparison of BEAM-F and baseline methods on macro F-measure as a function of (b) the amount of class skewness; (c) the mean separation parameter β between the classes.

Table I
STATISTICS OF THE REAL-WORLD BENCHMARK DATA SETS USED IN THE EXPERIMENTS. SKEW SIGNIFIES THE CLASS IMBALANCE I.E. $\frac{\max_i p_i}{\min_i p_i}$.

Data set	# instances	# features	# classes	Skew
connect-4	67557	126	3	6.9
covtype	581012	54	7	103.1
dna	2000	180	3	2.3
nursery	12958	27	4	13.1
poker	25010	10	10	2498.6
shuttle	43500	9	7	5684.7
usps	7291	256	10	2.2
MACHO	6059	64	8	67.2

Table II
MACRO F-MEASURE PERFORMANCE OF DIFFERENT METHODS. BEAM-F GIVES THE BEST OR COMPARABLE MACRO F-MEASURE PERFORMANCE, OFTEN BEATING BASELINES BY A WIDE MARGIN.

Data set	BEAM-F	LR	Bal-LR	CS-SVM
connect-4	**0.570**	0.499	**0.570**	0.498
covtype	**0.565**	0.487	0.476	0.488
dna	**0.945**	0.943	0.942	0.939
nursery	**0.901**	0.696	0.873	0.888
poker	**0.096**	0.067	0.020	0.059
shuttle	0.460	0.461	0.340	**0.625**
usps	0.948	0.948	0.943	**0.952**
MACHO	**0.605**	0.314	0.359	0.316

and BEAM-F were tuned using 3-fold cross-validation on the training set. In BEAM-F, the complete training set was used for both learning a class probability model, and for running the alternating maximization solver.

Real-world Benchmark Data. We evaluated BEAM-F on a variety of real-world benchmark data sets with varying degree of class skewness (see Table I). The results are tabulated in Table II. It is notable that BEAM-F is the winner or tied winner on most datasets. However, BEAM-F offers a significant improvement over the Bal-LogReg method whenever the skew of the dataset is high. This is to be expected since, on excessively skewed datasets, the Bal-LogReg method places a huge weight on the rare classes, distorting its view of the data. This trend will be corroborated in the synthetic experiments. Also, notice that on these highly skewed datasets, the simple LogReg performs better, as it does not encounter these huge weights. However, BEAM-F mostly beats both these methods and CS-SVM. It is also worth noting that the behaviour of CS-SVM is very erratic as it does not seek to optimize macro F-measure at all. On the other hand, Table III confirms that BEAM-F is much more scalable than the CS-SVM method, being as much as $50\times$ faster.

Synthetic Data. To closely analyze the behavior of various algorithms, we considered a toy 3-class classification problem with 2-dimensional features (see Figure 1(a)). The data distributions of the 3 classes was conditionally Gaussian with common covariance and means $\mu_i, i = 1, 2, 3$ chosen so that class 2 is sandwiched between the other two classes.

In the first experiment, we varied p_2 from 0.1 to 0.9 and set $p_1 = p_3 = 0.5 - p_2/2$, thereby increasing the class skew p_2/p_1. Figure 1(b) reveals two clear trends. Firstly, for extremely high skew, LogReg outperforms the more careful Bal-LogReg method, a trend we observed in the benchmark datasets as well. Secondly, BEAM-F consistently outperforms all methods, more so when skew is larger.

In the second experiment, we fixed class proportions at $p_1 = 0.05, p_2 = 0.1$, and $p_3 = 0.85$, but set the means to $\mu_1 = -\mu 3 = [-\beta, -\beta]^\top$, $\mu_2 = [0, 0]^\top$, where β was varied from 0.1 to 10. A larger β produces a simpler problem with large class separations. Figure 1(c) confirms that BEAM-F is consistently the best method, especially when the classes are closely spaced and the problem is hard to solve. As the separation increases and the problem becomes easier, other methods improve in performance as well.

Case Study: Celestial Object Classification. As a case study, we applied BEAM-F to the task of classifying celestial objects from the Massive Compact Halo Object (MACHO) catalog using photometric time series data [20]. There are 6059 light curves corresponding to either one of seven celestial objects or a miscellaneous category. The dataset is highly skewed, making it an ideal candidate for macro F-measure optimization. As Table II indicates, BEAM-F method yields a significantly higher macro F-measure compared to all other baselines.

Table III
TRAINING TIMES (SECS) OF BEAM-F AND CS-SVM. BEAM-F CAN BE SEVERAL TIMES FASTER THAN THE CS-SVM SOLVER.

Data set	BEAM-F	CS-SVM	Speedup
connect-4	17.9	79.1	4.4×
covtype	207.1	354.2	1.7×
poker	10.2	496.2	48×
shuttle	10.8	21.9	2×

	1	2	3	4	5	6	7	8
1	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00
4	0.00	0.00	0.00	0.01	0.00	0.05	0.00	0.00
5	0.00	0.00	0.00	0.00	0.02	0.07	0.00	0.00
6	0.00	0.00	0.00	0.00	0.02	**0.65**	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00

(a) LogReg

	1	2	3	4	5	6	7	8
1	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02
4	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.05	0.02	0.02	0.00
6	0.00	0.00	0.00	0.00	0.01	**0.60**	0.02	0.04
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.05

(b) BEAM-F

Figure 2. Confusion Matrices for the Celestial Objective Detection task. BEAM-F does not neglect rare classes in favor of the majority class.

The confusion matrices (see Figure 2) of the LogReg and BEAM-F method provide greater insight into the results. Notice that LogReg performs very well on class 6 while sacrificing performance on most other classes. On the other hand, BEAM-F yields non-zero accuracies on all but one class, at the cost of lower accuracy on class 6.

Acknowledgements. PK thanks the Deep Singh and Daljeet Kaur Faculty Fellowship, and the Research-I Foundation at IIT Kanpur for support.

REFERENCES

[1] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworth-Heinemann, 1979.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[3] N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu, "Optimizing F-Measures: A Tale of Two Approaches," in *29th International Conference on Machine Learning*, 2012.

[4] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent Binary Classification with Generalized Performance Metrics," in *28th Annual Conference on Neural Information Processing Systems*, 2014.

[5] S. P. Parambath, N. Usunier, and Y. Grandvalet, "Optimizing F-Measures by cost-sensitive classification," in *28th Annual Conference on Neural Information Processing Systems*, 2014.

[6] H. Narasimhan, R. Vaish, and S. Agarwal, "On the Statistical Consistency of Plug-in Classifiers for Non-decomposable Performance Measures," in *28th Annual Conference on Neural Information Processing Systems*, 2014.

[7] T. Joachims, "A Support Vector Method for Multivariate Performance Measures," in *22nd International Conference on Machine Learning*, 2005.

[8] R. Busa-Fekete, B. Szörényi, K. Dembczynski, and E. Hüllermeier, "Online F-Measure Optimization," in *29th Annual Conference on Neural Information Processing Systems*, 2015.

[9] H. Narasimhan, P. Kar, and P. Jain, "Optimizing Non-decomposable Performance Measures: A Tale of Two Classes," in *32nd International Conference on Machine Learning*, 2015.

[10] H. Narasimhan, H. G. Ramaswamy, A. Saha, and S. Agarwal, "Consistent Multiclass Algorithms for Complex Performance Measures," in *32nd International Conference on Machine Learning*, 2015.

[11] J. Petterson and T. Caetano, "Reverse Multi-Label Learning," in *Advances in Neural Information Processing Systems*, 2010.

[12] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, "An Exact Algorithm for F-Measure Maximization," in *25th Annual Conference on Neural Information Processing Systems*, 2011.

[13] K. Dembczyński, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier, "Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization," in *30th International Conference on Machine Learning*, 2013.

[14] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent Multilabel Classification," in *29th Annual Conference on Neural Information Processing Systems*, 2015.

[15] S. Schaible and J. Shi, "Fractional programming: The sum-of-ratios case," *Optimization Methods and Software*, vol. 18, no. 2, pp. 219–229, 2003.

[16] H. P. Benson, "On the Global Optimization of Sums of Linear Fractional Functions over a Convex Set," *Journal of Optimization Theory and Applications*, vol. 121, no. 1, pp. 19–39, 2004.

[17] M. Jaggi, "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization," in *30th International Conference on Machine Learning*, 2013.

[18] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, Belmont MA, 1999.

[19] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex Sets and Optimization with Biconvex Functions - A Survey and Extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[20] C. Alcock *et al.*, "The MACHO Project: Microlensing Results from 5.7 Years of LMC Observations," *The Astrophysical Journal*, vol. 542, no. 1, p. 281, 2000.