PD-Sparse : A Primal and Dual Sparse Approach to Extreme Multiclass and Multilabel Classification

Ian E. H. Yen [1 *] IANYEN@CS.UTEXAS.EDU
Xiangru Huang [1 *] XRHUANG@CS.UTEXAS.EDU
Kai Zhong [2] ZHONGKAI@ICES.UTEXAS.EDU
Pradeep Ravikumar [1,2] PRADEEPR@CS.UTEXAS.EDU
Inderjit S. Dhillon [1,2] INDERJIT@CS.UTEXAS.EDU

[*] Both authors contributed equally.

[1] Department of Computer Science, University of Texas at Austin, TX 78712, USA.
[2] Institute for Computational Engineering and Sciences, University of Texas at Austin, TX 78712, USA.

Abstract

We consider multiclass and multilabel classification with extremely large number of classes. In such a setting, standard methods that have training, prediction cost linear to the total number of classes become intractable. State-of-the-art methods thus reduce complexity via structural assumption between labels such as low rankness or balanced tree. However, in practice, structural assumption can be easily violated, which leads to degradation in the prediction accuracy. In this paper, we show that a margin-maximizing loss with L1-penalty, in the case of Extreme Classification, yields an extremely sparse solution both in primal and in dual without sacrificing the expressive power of predictor. We thus propose a variant of Block-Coordinate Frank-Wolfe algorithm that exploits the sparsity to achieve complexity sublinear to number of primal and dual variables. In our experiments on Extreme multiclass and multilabel problems, the proposed method achieves significantly higher accuracy than existing approaches with very competitive training and prediction time.

1. Introduction

Extreme Classification encompasses multiclass and multilabel problems with huge number of classes or labels. Problems of this kind are prevalent in real-world applications such as text, image or video annotation, where one aims to learn a predictor that tags a data point with the most relevant labels out of a huge collection. In the multiclass setting, we are given the fact that only one label is correct, while in the multilabel setting, multiple labels are allowed.

In the Extreme Classification setting, standard approaches such as one-versus-all and one-versus-one become intractable both in training and prediction phase due to computations involving large number of model parameters (Deng et al., 2010). Recently several approaches have been proposed to exploit structural relations between labels for reducing training and prediction time. A natural approach is to find an embedding so the model parameters of each label can be projected to a low-dimensional space, reducing the cost in training and prediction (Chen & Lin, 2012; Yu et al., 2013; Kapoor et al., 2012). However, in real applications, the data may not be low-rank and the low-rank approaches may result in lower accuracy. Furthermore, for high-dimensional data with a sparse feature matrix, the model learned from low-rank approach can project a sparse feature vector into a dense vector that results in even higher prediction cost than a simple linear classifer.

Another recent thread of research has investigated tree-based methods that partition labels into tree-structured groups, so in both training and prediction phases, one can follow tree branches to avoid accessing irrelevant models (Prabhu & Varma, 2014; Choromanska & Langford, 2015; Choromanska et al., 2013). However, finding a balanced tree structure that partitions labels effectively in the feature space is a difficult problem in itself. While many heuristics have been proposed for finding a good tree structure, in practice, one needs to ensemble several trees to achieve performance comparable to standard classifiers.

In this paper, instead of making a structural assumption on the label relationships, we assume that for each instance,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

there are only a few correct labels and the feature space is rich enough for one to distinguish between labels. This assumption is much weaker than other structural assumptions, and under such an assumption, we show that a simple margin-maximizing loss yields an extremely sparse dual solution in the setting of extreme classification. Furthermore, the loss, when combined with ℓ_1 penalty, gives a sparse solution both in the primal and in the dual for any ℓ_1 regularization parameter $\lambda > 0$.

We thus propose a Fully-Corrective Block Coordinate Frank-Wolfe algorithm to solve the primal-dual sparse problem given by margin-maximizing loss with ℓ_1-ℓ_2 penalties. Let D be the problem dimension, N be the number of samples, and K be the number of classes. In case $DK \gg N$, the proposed algorithm has complexity sublinear to the number of variables by exploiting sparsity in the primal to search active variables in the dual. In case $DK \lesssim N$, we propose a stochastic approximation method to further speed up the search step in the Frank-Wolfe algorithm.

In our experiments on both multiclass and multilabel problems, the proposed method achieves significantly higher accuracy than existing approaches of Extreme Classification with competitive training and prediction time.

2. Problem Formulation

Our formulation is based on the Empirical Risk Minimization (ERM) framework. Given a collection of training instances $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ is D-dimensional (possibly sparse) feature vector of i-th instance and $\boldsymbol{y}_i \in \{0,1\}^K$ is label indicator vector with $y_{ik} = 1$ if k is a correct label for the i-th instance and $y_{ik} = 0$ otherwise. We will use $\mathcal{P}(\boldsymbol{y}) = \{k \in [K] \mid y_k = 1\}$ to denote positive label indexes, while using $\mathcal{N}(\boldsymbol{y}) = \{k \in [K] \mid y_k = 0\}$ to denote the negative label indexes. In this work, we assume the number of labels K is extremely large but the number of positive labels $nnz(\boldsymbol{y})$ is small and not growing linearly with K. For example, in multiclass classification problem, we have $nnz(\boldsymbol{y}) = 1$, and the assumption is also satisfied typically in mulitlabel problems. Denote $X := (\boldsymbol{x}_i^T)_{i=1}^N$ as the $N \times D$ design matrix and $Y := (\boldsymbol{y}_i^T)_{i=1}^N$ as the N by K label matrix, our goal is to learn a classifier $h : \mathbb{R}^D \rightarrow [K]$

$$h(\boldsymbol{x}) := \underset{k}{argmax}\ \langle \boldsymbol{w}_k, \boldsymbol{x} \rangle, \tag{1}$$

parameterized by a $D \times K$ matrix $W = (\boldsymbol{w}_k)_{k=1}^K$.

Loss with Dual Sparsity In this paper, we consider the *separation ranking loss* (Crammer & Singer, 2003) that penalizes the prediction on an instance $\boldsymbol{x}$ by the highest response from the set of negative labels minus the lowest response from the set of positive labels

$$L(\boldsymbol{z}, \boldsymbol{y}) = \max_{k_n \in \mathcal{N}(\boldsymbol{y})} \max_{k_p \in \mathcal{P}(\boldsymbol{y})} \left(1 + z_{k_n} - z_{k_p}\right)_+ \tag{2}$$

where an instance has zero loss if all positive labels $k_p \in \mathcal{P}_i$ have higher responses than that of negative labels $k_n \in \mathcal{N}_i$ plus a margin. In the multiclass setting, let $p(\boldsymbol{y})$ be the unique positive label. The loss (2) becomes the well-known multiclass SVM loss

$$L(\boldsymbol{z}, \boldsymbol{y}) = \max_{k \in [K] \setminus \{p(\boldsymbol{y})\}} \left(1 + z_k - z_{p(\boldsymbol{y})}\right)_+ \tag{3}$$

proposed in (Crammer & Singer, 2002) and widely-used in linear classification package such as LIBLINEAR (Fan et al., 2008). The basic philosophy of loss (2) is that, for each instance, there are only few labels with high responses, so one can boost prediction accuracy by learning how to distinguish between those confusing labels. Note the assumption is reasonable in Extreme Classification setting where K is large and only few of them are supposed to give high response. On the other hand, this does not give much computational advantage in practice, since, to identify labels of high response for each instances, one still needs to evaluate (1) for $\forall n \in [N], \forall k \in [K]$, resulting in an $O(nnz(X)K)$ complexity that is of the same order to the one-vs-all approach. (Keerthi et al., 2008) proposed an approach in the multiclass setting that tries to identify active variables corresponding to labels of high responses in the dual formulation of the ℓ_2-regularized instance

$$\frac{1}{2} \sum_{k=1}^K \|\boldsymbol{w}_k\|^2 + C \sum_{i=1}^N L(W^T \boldsymbol{x}_i, y_i), \tag{4}$$

where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_K]$. Let $\boldsymbol{\alpha}_k := (\alpha_{ik})_{i \in [N]}$ and $\boldsymbol{\alpha}^i := (\alpha_{ik})_{k \in [K]}$. The dual problem is of the form

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{k=1}^K \|\boldsymbol{w}_k(\boldsymbol{\alpha})\|^2 + \sum_{i=1}^N \boldsymbol{e}^T \boldsymbol{\alpha}^i \\ s.t. \quad & \boldsymbol{\alpha}^i \in \Delta_i^K, \ \forall i \in [N] \end{aligned} \tag{5}$$

where

$$\boldsymbol{w}_k(\boldsymbol{\alpha}_k) = \sum_{i=1}^N \alpha_{ik} \boldsymbol{x}_i = X^T \boldsymbol{\alpha}_k, \tag{6}$$

$\boldsymbol{e}_i = \boldsymbol{1} - \boldsymbol{y}_i$, and $\Delta_i^K = \{\boldsymbol{\alpha} \mid \sum_{k=1}^K \alpha_k = 0,\ \alpha_{p(\boldsymbol{y}_i)} \leq C,\ \alpha_k \leq 0,\ \forall k \neq p(\boldsymbol{y}_i)\}$ is a shifted simplex of K corners. In particular, the optimal solution $\boldsymbol{\alpha}$ of (5) satisfies: for $k \neq p(\boldsymbol{y}_i)$, $\alpha_{ik}^* \neq 0$ if and only if label k has highest response $z_{ik} = \langle \boldsymbol{w}_k, \boldsymbol{x}_i \rangle$ that attains the maximum of (3). Therefore, to identify active variables that correspond to the confusing labels, (Keerthi et al., 2008) proposes a *shrinking* heuristic that "shrinks" a dual variable whenever its descent direction towards the boundary. The shrunken variables are then excluded from the optimization, which in practice reduces training time by orders of magnitude. While the shrinking heuristic is quite successful for problem of medium number of class K. For problem of K more than 10^4 labels, the technique becomes impractical since even computing gradient for each of the $N \times K$ variables once requires days of time and hundreds of gigabytes of memory (as shown in our experiments).

Primal and Dual-Sparse Formulation One important observation that motivates this work is the intriguing property of ERM with dual-sparse loss (2) and ℓ_1 penalty

$$\lambda \sum_{k=1}^{K} \|\boldsymbol{w}_k\|_1 + \sum_{i=1}^{N} L(W^T \boldsymbol{x}_i, \boldsymbol{y}_i). \tag{7}$$

in the setting of Extreme Classification. Consider the optimal solution W^* of (7), which satisfies

$$\lambda \boldsymbol{\rho}_k^* + \sum_{i=1}^{N} \alpha_{ik}^* \boldsymbol{x}_i = \lambda \boldsymbol{\rho}_k^* + X^T \boldsymbol{\alpha}_k = 0, \ \forall k \in [K] \tag{8}$$

for some subgradients $\boldsymbol{\rho}_k^* \in \partial \|\boldsymbol{w}_k^*\|_1$ and $\boldsymbol{\alpha}^{i*} \in \partial_{\boldsymbol{z}} L(\boldsymbol{z}_i, \boldsymbol{y}_i)$ with $\boldsymbol{z}_i = W^{*T} \boldsymbol{x}_i$. Recall that the subgradients $\boldsymbol{\alpha}^i$ of loss (2) have $\alpha_{ik^*} \neq 0$ for some $k^* \neq \bar{k}$ only if k^* is the confusing label that satisfies

$$k^* \in arg \max_{k \neq \bar{k}} \langle \boldsymbol{w}_k, \boldsymbol{x}_i \rangle.$$

This means we have $nnz(\boldsymbol{\alpha}^i) \ll K$ and $nnz(A) \ll NK$ as long as there are few labels with higher responses than the others, which is satisfied in most of Extreme Classification problems. On the other hand, the subgradient $\boldsymbol{\rho}_k$ of ℓ_1-norm satisfies

$$\rho_{jk} = \begin{cases} 1, & w_{jk}^* > 0 \\ -1, & w_{jk}^* < 0 \\ \nu, & \nu \in [-1, 1], w_{jk}^* = 0, \end{cases} \tag{9}$$

which means the set of non-zero primal variables $\mathcal{B}_k^* = \{j \mid w_{jk}^* \neq 0\}$ at optimal satisfies

$$\lambda \, sign([\boldsymbol{w}_k^*]_{\mathcal{B}_k^*}) \mathbf{1}_{\mathcal{B}_k^*} = \left[X^T \boldsymbol{\alpha}_k^* \right]_{\mathcal{B}_k^*}, \tag{10}$$

which is a linear system of $|\mathcal{B}_k^*|$ equality constraints and $nnz(\alpha_k)$ variables. However, for general design matrix X that draws from any continuous probability distribution (Tibshirani et al., 2013), the above cannot be satisfied unless

$$nnz(\boldsymbol{w}_k^*) = |\mathcal{B}_k^*| \leq nnz(\boldsymbol{\alpha}_k^*), \forall k \in [K] \tag{11}$$

and (11) further implies

$$nnz(W^*) \leq nnz(A^*) \tag{12}$$

by summation over K, where A^* an $N \times K$ matrix of stacked $(\boldsymbol{\alpha}_k^*)_{k=1}^K$. This means in Extreme Classification problem, not only non-zero dual variables but also primal variables are sparse at optimal. Note this result holds for any ℓ_1 parameter $\lambda > 0$, that means it does not gain primal sparsity via sacrificing the expressive power of the predictor. Instead, it implies there exists a naturally sparse optimal solution W^* under the loss (2), which can be found through imposing a very small ℓ_1 penalty. The result is actually a simple extension to the fact that the number of nonzero weights at optimal under ℓ_1 penalty is less or equal to the number of samples (Tibshirani et al., 2013). We summarize the result as following Corollary.

Corollary 1 (Primal and Dual Sparsity)**.** *The optimal primal and dual solution* (W^*, A^*) *of ERM problem* (7) *with loss* (2) *satisfies*

$$nnz(W^*) \leq nnz(A^*)$$

for any $\lambda > 0$ *if the design matrix* X *is drawn from a continuous probability distribution.*

Dual Optimization via Elastic Net Although (7) has superior sparsity, both the primal and dual optimization problems for (7) are non-smooth and non-separable w.r.t. coordinates, where greedy coordinate-wise optimization could be non-convergent [1]. However, from the duality between strong convexity and dual smoothness (Kakade et al., 2009; Meshi et al., 2015), this issue can be resolved simply via adding an additional strongly convex term in the primal. In particular, by adding an ℓ_2 regularizer to (7), the *Elastic-Net*-regularized problem

$$\sum_{k=1}^{K} \frac{1}{2} \|\boldsymbol{w}_k\|^2 + \lambda \|\boldsymbol{w}_k\|_1 + C \sum_{i=1}^{N} L(W^T \boldsymbol{x}_i, \boldsymbol{y}_i) \tag{13}$$

has dual form

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & G(\boldsymbol{\alpha}) := \frac{1}{2} \sum_{k=1}^{K} \|\boldsymbol{w}_k(\boldsymbol{\alpha}_k)\|^2 + \sum_{i=1}^{N} \boldsymbol{e}_i^T \boldsymbol{\alpha}^i \\ s.t. \quad & \boldsymbol{\alpha}^i \in \mathcal{C}_i, \ \forall i \in [N] \end{aligned} \tag{14}$$

that entangles variable of different samples $\boldsymbol{\alpha}^i$, $\boldsymbol{\alpha}^{i'}$ only through a smooth term $\sum_{k=1}^{K} \|\boldsymbol{w}_k(\boldsymbol{\alpha}_k)\|^2/2$, where

$$\boldsymbol{w}_k(\boldsymbol{\alpha}_k) := \mathbf{prox}_{\lambda \|.\|_1}(X^T \boldsymbol{\alpha}_k). \tag{15}$$

and

$$\mathcal{C}_i := \left\{ \boldsymbol{\alpha} \middle| \begin{array}{l} \sum_{k \in \mathcal{N}_i} (-\alpha_k) = \sum_{k \in \mathcal{P}_i} \alpha_k \in [0, C], \\ 0 \leq \alpha_k, \forall k \in \mathcal{P}_i, \ \alpha_k \leq 0, \forall k \in \mathcal{N}_i \end{array} \right\}. \tag{16}$$

The proximal operator of ℓ_1-norm $\mathbf{prox}_{\lambda \|.\|_1}(\boldsymbol{v})$ performs soft-thresholding to each single element v_j as

$$\mathbf{prox}_{\lambda |.|}(v_j) := \begin{cases} 0, & |v_j| \leq \lambda \\ v_j - \lambda, & v_j > \lambda \\ v_j + \lambda, & v_j < \lambda \end{cases}$$

The dual problem (14) has very similar form to that from purely ℓ_2 regularized problem (5), with difference on the definition of $\boldsymbol{w}_k$ (15), where the ℓ_1-ℓ_2-regularized problem has $\boldsymbol{w}_k(\boldsymbol{\alpha}_k)$ being a sparse vector obtained from applying soft-thresholding operator to $X^T\boldsymbol{\alpha}_k$. This, however, leads to the key to our efficiency gain. In particular, the objective allows efficient search of active dual variables via sparsity in the primal, while allows efficient maintenance of nonzero primal variables through an active-set strategy in the dual.

Note the Elastic-Net-regularized problem could not satisfy corollary 1. However, empirically, it has been observed to produce solution of sparsity close to that from ℓ_1 regularizer, while the solution from Elastic-Net is often of higher prediction accuracy (Zou & Hastie, 2005). In our experiments, we have observed extremely sparse primal solution from (13) which not only help in the training phase but also results in faster prediction that is competitive to the logarithmic-time prediction given by tree-based approach (Choromanska & Langford, 2015).

3. Algorithm

The objective (14) comprises a smooth function subject to constraints $\mathcal{C}_1, .., \mathcal{C}_N$ separable w.r.t. blocks of variables $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_N$. A fast convergent algorithm thus minimizes (14) one block at a time. In this section, we propose a Fully-Corrective Block-Coordinate Frank-Wolfe (BCFW) for the dual problem (14) that explicitly taking advantage of the primal and dual sparsity.

Note for the similar dual problem (5) resulted from L2-regularization, a BCFW method that searches the greedy coordinate α^*_{ik} at each iterate is not better than a Block Coordinate Descent (BCD) algorithm that performs updates on the whole block of variable $\boldsymbol{\alpha}^i$ (Keerthi et al., 2008; Fan et al., 2008), since the greedy search requires evaluation of gradient w.r.t. each coordinate, which results in the same cost to minimizing the whole block of variables, given the minimization can be done via a simplex projection.

On the other hand, our dual objective (14) has gradient of i-th block equals

$$\nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}) = W^T\boldsymbol{x}_i - \boldsymbol{e}_i. \tag{17}$$

If a primal-sparse W can be maintained via (15), the gradient can be evaluated in time $O(nnz(x_i)nnz(w_j))$ (and $O(nnz(W))$ for dense x_i). In contrast, the update of the whole block of variable $\boldsymbol{\alpha}^i$ would require maintaining relation (15) for $\boldsymbol{w}_1...\boldsymbol{w}_K$, which cannot exploit sparsity of $\boldsymbol{w}_k$ and would require $O(nnz(x_i)K)$ (and $O(DK)$ for dense x_i). So in the Extreme Classification setting, the cost of updating an coordinate is orders of magnitude larger than cost of evaluating its gradient.

Algorithm 1 Fully-Corrective BCFW

```
Initialize $\boldsymbol{\alpha}^0 = \mathbf{0}$, $\mathcal{A}^0 = \emptyset$.
for $t = 1...T$ do
    Draw a sample index $i \in [N]$ uniformly at random.
    Find most-violating label $k^* \in \mathcal{N}_i$ via (20).
    $\mathcal{A}_i^{t+\frac{1}{2}} = \mathcal{A}_i^t \cup \{k^*\}$.
    Solving subproblem (21) w.r.t. active set $\mathcal{A}_i^{t+\frac{1}{2}}$.
    $\mathcal{A}_i^{t+1} = \mathcal{A}_i^{t+\frac{1}{2}} \setminus \{k \mid \alpha_{ik} = 0, k \notin \mathcal{P}_i\}$.
    Maintain $\boldsymbol{w}(\boldsymbol{\alpha})$, $\boldsymbol{v}(\boldsymbol{\alpha})$ via (23).
end for.
```

3.1. Fully-Corrective Block-Coordinate Frank Wolfe (FC-BCFW)

As a result, we employ a BCFW strategy where the updates of variables are restricted to an active set of labels $\mathcal{A}_i^t$ for each sample i. In each iteration, the BCFW method draws a block of variables $\boldsymbol{\alpha}^i$ uniformly from $\{\boldsymbol{\alpha}^i\}_{i=1}^N$, and finds greedy direction based on a local linear approximation

$$\boldsymbol{\alpha}^{it}_{FW} := \underset{\boldsymbol{\alpha}^i \in \mathcal{C}_i}{argmin} \ \langle \nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^i \rangle. \tag{18}$$

For $\mathcal{C}_i$ of structure (16), (18) is equivalent to finding the most violating pair of positive, negative labels:

$$(k_n^*, k_p^*) := \underset{k_n \in \mathcal{N}_i, k_p \in \mathcal{P}_i}{argmin} \ \langle \nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}^t), (\boldsymbol{\delta}_{k_p} - \boldsymbol{\delta}_{k_n}) \rangle, \tag{19}$$

where $\boldsymbol{\delta}_k$ is $K \times 1$ indicator vector for k-th variable. However, since we are considering problem where $|\mathcal{P}_i|$ is small, we can keep all positive labels in the active set $\mathcal{A}_i$. Then to guarantee that the FW direction (18) is considered in the active set, we only need to find the most-violating negative label:

$$\begin{aligned} k_n^* &:= \underset{k_n \in \mathcal{N}_i}{argmax} \ \langle \nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}^t), \boldsymbol{\delta}_{k_n} \rangle, \\ &= \underset{k_n \in \mathcal{N}_i}{argmax} \ \langle \boldsymbol{w}_k^t, \boldsymbol{x}_i \rangle - 1 \end{aligned} \tag{20}$$

which can be computed in $O(nnz(\boldsymbol{x}_i)nnz(\boldsymbol{w}_{\bar{j}}))$, where $\bar{j}$ is the feature j of most non-zero labels in W, among all nonzero features $\boldsymbol{x}_i$.

After adding k_n^* to the active set, we minimize objective (14) w.r.t. the active set and fix $\alpha_{ik} = 0$ for $\forall k \notin \mathcal{A}_i$. By noticing that the Hessian matrix $\nabla_{ii}^2 G$ w.r.t. a block of variables $\boldsymbol{\alpha}^i$ is a diagonal matrix of diagonal element

$$\nabla_{\alpha_{ik}}^2 G = [\boldsymbol{x}_i]_{\mathcal{B}_k}^T [\boldsymbol{x}_i]_{\mathcal{B}_k} \leq \|\boldsymbol{x}_i\|^2$$

where $[.]_{\mathcal{B}_k}$ denotes the sub-vector by taking indexes belonging to the primal non-zero variables of k-th label, one

[1] The coordinate descent method has global convergence only on problem where the non-smooth terms are separable w.r.t. the coordinates.

Algorithm 2 Projection for Subproblem (21)

```
$\min_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \|\boldsymbol{y}-\boldsymbol{c}\|_2^2$
s.t. $\boldsymbol{x} \succeq 0, \boldsymbol{y} \succeq 0$, $\|\boldsymbol{x}\|_1 = \|\boldsymbol{y}\|_1 = t \in [0, C]$
Also, WLOG assume $\boldsymbol{b}_0 \geq \ldots \geq \boldsymbol{b}_{N-1}, \boldsymbol{c}_0 \geq \ldots \geq \boldsymbol{c}_{M-1}$
$S_b(n) = \sum_{k=0}^{n} \boldsymbol{b}_k$, $S_c(m) = \sum_{k=0}^{m} \boldsymbol{c}_k$, $D_b(n) = S_b(n) - (n+1)\boldsymbol{b}_n$, $D_c(m) = S_c(m) - (m+1)\boldsymbol{c}_m$,
Use points in $\{D_b(n), D_c(m)\}$ to split interval $[0, C]$.
for each interval $I = [l, r]$ do
    $n_I = \max_{D_b(n) \leq l} n, m_I = \max_{D_c(m) \leq l} m$
    $t_I = \frac{(n_I+1)S_c(m_I)+(m_I+1)S_b(n_I)}{(n_I+m_I+2)}$ clip to $[l, r]$
    $v_I = \sum_{k=n_I+1}^{N-1} \boldsymbol{b}_k^2 + \sum_{k=m_I+1}^{M-1} \boldsymbol{c}_k^2 + \frac{(S_b(n_I)-t_I)^2}{n_I+1} + \frac{(S_c(m_I)-t_I)^2}{m_I+1}$, update $I^* = arg\min_I v_I$.
end for
$\boldsymbol{x}_k = \begin{cases} 0 & \text{if } k > n_{I^*} \\ \boldsymbol{b}[k] + (t_{I^*} - S_b(n_{I^*})/(n_{I^*}+1) & \text{else} \end{cases}$
$\boldsymbol{y}_k = \begin{cases} 0 & \text{if } k > m_{I^*} \\ \boldsymbol{c}[k] + (t_{I^*} - S_c(m_{I^*})/(m_{I^*}+1) & \text{else} \end{cases}$
```

can use $\|\boldsymbol{x}_i\|^2$ as a simple upper bound on each diagonal element of the Hessian matrix, and solves the following block subproblem

$$\min_{\boldsymbol{\alpha}_{\mathcal{A}_i} \in \mathcal{C}_i} \langle \nabla_{\boldsymbol{\alpha}_{\mathcal{A}_i}} G, \boldsymbol{\alpha}_{\mathcal{A}_i} - \boldsymbol{\alpha}^t_{\mathcal{A}_i} \rangle + \frac{Q_i}{2} \|\boldsymbol{\alpha}_{\mathcal{A}_i} - \boldsymbol{\alpha}^t_{\mathcal{A}_i}\|^2 \tag{21}$$

where $Q_i = \|\boldsymbol{x}_i\|^2$ and $\mathcal{A}_i = \mathcal{A}_i^t \cup \{k_n^*\}$. Note, when $|\mathcal{P}_i| = 1$, the subproblem (21) can be solved by a simple projection to simplex of complexity $O(|\mathcal{A}_i| \log |\mathcal{A}_i|)$. For $|\mathcal{P}_i| > 1$, we derive a similar procedure that generalizes projection of simplex to that for the constraint $\mathcal{C}_i$ of the same complexity. The problem can be expressed as (22), and solved as depicted in Algorithm 2.

$$\begin{aligned} \min_{\boldsymbol{\alpha}_{\mathcal{A}_i} \in \mathcal{C}_i} & \|(-\boldsymbol{\alpha}^{\mathcal{N}_i}_{\mathcal{A}_i}) - \boldsymbol{b}\|_2^2 + \|\boldsymbol{\alpha}^{\mathcal{P}_i}_{\mathcal{A}_i} - \boldsymbol{c}\|_2^2 \\ \text{s.t. } & \boldsymbol{\alpha}^{\mathcal{P}_i}_{\mathcal{A}_i}, \boldsymbol{\alpha}^{\mathcal{N}_i}_{\mathcal{A}_i} \text{ is a partition of } \boldsymbol{\alpha}_{\mathcal{A}_i} \text{ w.r.t. } \mathcal{P}_i, \mathcal{N}_i \\ & \boldsymbol{b}_k = (\langle \boldsymbol{w}_k^t, \boldsymbol{x}_i \rangle + 1)/Q_i - \boldsymbol{\alpha}^t_{\mathcal{A}_i}(k), \forall k \in \mathcal{N}_i \\ & \boldsymbol{c}_k = \boldsymbol{\alpha}^t_{\mathcal{A}_i}(k) - \langle \boldsymbol{w}_k^t, \boldsymbol{x}_i \rangle / Q_i, \forall k \in \mathcal{P}_i \end{aligned} \tag{22}$$

After solving the subproblem (21) w.r.t. active set $\mathcal{A}_i$, we update $\boldsymbol{w}_k(\boldsymbol{\alpha}_k^t)$ to $\boldsymbol{w}_k(\boldsymbol{\alpha}_k^{t+1})$ by maintaining an additional vector $\boldsymbol{v}_k^t$ such that

$$\boldsymbol{v}_k^t = X^T \boldsymbol{\alpha}_k^t, \quad \boldsymbol{w}_k^t = \mathbf{prox}_{\lambda\|.\|}(\boldsymbol{v}_k^t). \tag{23}$$

where maintaining the first relation costs $O(nnz(\boldsymbol{x}_i)|\mathcal{A}_i|)$ and maintaining the second requires the same cost by checking only values changed by the first step. Note to evaluate (20) efficiently for $k \in \mathcal{N}_i$, one needs to maintain not only $\boldsymbol{w}_k$ that supports random access but also a list of nonzero indexes for each feature $j \in [D]$, which can be maintained as a hashed set in order to be updated efficiently in (23) and also traversed efficiently in (20). We will describe how to design an optimized data structure for this task in section 4.2.

The procedure is summarized in Algorithm 1. Note in practice, we use *sampling without replacement* instead of *uniform sampling with replacement* at step 1 of algorithm 1 that ensures (i) every sample is updated at least once and (ii) the size of active set $|\mathcal{A}_i|$ is bounded by the number of BCFW pass.

The overall space requirement of Algorithm 1 for storing non-zero dual variables $\{\boldsymbol{\alpha}^i\}_{i=1}^N$ is bounded by $|\mathcal{A}| \ll NK$, while the storage for maintaining primal variable is dominated by the space for $\{\boldsymbol{v}_k\}_{k=1}^K$, which in the worst case, requires $O(DK)$. However, by the definition of $\boldsymbol{v}_k$ (23), the number of non-zero elements in $\{\boldsymbol{v}_k\}_{k=1}^K$ is bounded by $O(nnz(X)\max_i(|\mathcal{A}_i|))$, with $\max_i(|\mathcal{A}_i|)$ bounded by the number of BCFW passes. This means the space requirement of the algorithm is only t times of the data size $nnz(X)$ for running t iterations. In practice, $|\mathcal{A}_i|$ converges to the number of active labels of sample i and does not increase after certain number of iterations.

The overall complexity for each iterate of FC-BCFW is $O(nnz(\boldsymbol{x}_i)nnz(\boldsymbol{w}_j^t) + nnz(\boldsymbol{x}_i)|\mathcal{A}_i^t|)$. In case data matrix is dense the cost for one pass of FC-BCFW over all variables can be written as $O(Nnnz(W) + Dnnz(A))$, where A is the N by K matrix reshape of $\boldsymbol{\alpha}$. Let

$$k_W := nnz(W)/D \ , \ k_A := nnz(A)/N$$

be the average number of active labels per feature and per sample respectively. We have

$$O(Nnnz(W) + Dnnz(A)) = O(NDk_W + NDk_A).$$

Note k_A is bounded by the number of BCFW passes, and it is generally small when label has diverse responses on each instance. On the other hand, suppose the Elastic-Net penalty leads to sparsity similar to that of ℓ_1-regularized problem (which we observed empirically). We have $nnz(W) \lesssim nnz(A)$ and thus $k_W \lesssim \frac{N}{D} k_A$, which means k_W is small if $D \approx N$. On the other hand, for problem of small dimension, the bound becomes useless as the $\frac{N}{D} k_A$ can be even larger than K. In such case, the search (20) becomes bottleneck. In the next section, we propose a stochastic approximation technique in section 3.2 to further speed up the greedy search step (20).

3.2. Bi-stochastic Approximate Greedy Search

In this section, we consider a bi-stocahstic faster alternative to the direction computation of (20) when $k_W >> k_A$, that is, when the search step (20) a bottleneck of iterate.

The first approximation scheme divides K into ν mutually exclusive subsets $[K] = \bigcup_{q=1}^{\nu} \mathcal{Y}^{(q)}$, and realizes an approximate greedy search by first sampling q uniformly from $[\nu]$ and returning

$$\hat{k} = arg \max_{k \in \mathcal{Y}^{(q)}} \langle \boldsymbol{w}_k^t, \boldsymbol{x}_i \rangle. \tag{24}$$

as the approximate Frank-Wolfe direction. Note there is at least $1/\nu$ probability a partition $\mathcal{Y}^{(q)}$ containing label k^* is drawn, so the approximation scheme guarantees a multiplicative approximation factor on the expected progress. In section 3.3, we show that Algorithm 1 has convergence to the optimum under approximation (24), with a rate scaled by $1/\nu$. Since (24) reduces the time spent on search by a factor of $1/\nu$, the worst-case analysis gives the same overall complexity. In practice (24) reduces training time significantly since the active set $\mathcal{A}_i$ typically identifies all the labels that are stably active after a few iterates.

Another approximation scheme that can be used jointly with (24) is *importance sampling* on the non-zero feature indexes of $\boldsymbol{x}_i$. Let $d_i = nnz(x_i)$. We sample $\tilde{d}_i$ non-zero features with probability $|x_{ij}|/\|\boldsymbol{x}_i\|_1$ for j feature. Denote $\mathcal{D}_i$ as the set of feature indices obtained from sampling. We further approximate (24) via

$$\hat{k} = arg \max_{k \in \mathcal{Y}^{(q)}} \bar{C}_k(\mathcal{D}_i), \tag{25}$$

where $\bar{C}_k(\mathcal{D}_i) = \frac{\|\boldsymbol{x}_i\|_1}{\tilde{d}_i} \sum_{j \in \mathcal{D}_i} \boldsymbol{w}_{kj}^t sign(\boldsymbol{x}_{ij})$ is an unbiased estimator of $\langle \boldsymbol{w}_k^t, \boldsymbol{x}_i \rangle$. The scheme (25) reduces search time by another factor of $\tilde{d}_i/d_i$. In fact, with probability $1-\delta$, $|\bar{C}_k(\mathcal{D}_i) - \langle \boldsymbol{w}_k^t, \boldsymbol{x}_i \rangle| \leq \epsilon_d$ holds for all $k \in [K]$, given that

$$\tilde{d}_i \gtrsim d_i \frac{\|\boldsymbol{x}_i\|_\infty^2 R_w^2 \log(\frac{K}{\delta})}{\epsilon_d^2}, \tag{26}$$

where R_w^2 is an upper bound on $\sum_{j:\boldsymbol{x}_{ij} \neq 0} (\boldsymbol{w}_{kj}^t)^2$. One can find the proof of (26) in Appendix C. Based on (26), we will discuss how to choose $\tilde{d}_i$ in practice in section 4.4.

The effect of (25) is a $(1/\nu, \epsilon_d/\nu)$ multiplicative-additive approximation factor on the expected descent amount of the iterate.

3.3. Convergence

The following theorem gives convergence result of the FC-BCFW Algorithm. The analysis follows the convergence proof in (Lacoste-Julien et al., 2013). We provide details in Appendix A.

Theorem 1 (Convergence of FC-BCFW)**.** *Let $G(\boldsymbol{\alpha})$ be the dual objective (14). The iterates $\{\boldsymbol{\alpha}^t\}_{t=1}^{\infty}$ given by the Fully-Corrective Block-Coordinate Frank-Wolfe (Algorithm 1) has*

$$G(\boldsymbol{\alpha}^t) - G^* \leq \frac{2(QR^2 + \Delta G^0)}{t/N + 2} \ , \ t \geq 0 \tag{27}$$

where $Q = \sum_{i=1}^{N} Q_i$, $\Delta G^0 := G(\boldsymbol{\alpha}^0) - G^*$ and $R = 2C$ is the diameter of the domain (16).

Note our objective $G(\boldsymbol{\alpha}^t)$ is N times of the objective defined by *average loss* in for example (Lacoste-Julien et al., 2013; Shalev-Shwartz & Zhang, 2013), so one would divide both sides of (27) by N to compare the rates. The following gives convergence of FC-BCFW when sampling approximation (25) is used.

Theorem 2 (Convergence with Sampling Approximation)**.** *Let $G(\boldsymbol{\alpha})$ be the dual objective (14). The iterates $\{\boldsymbol{\alpha}^t\}_{t=1}^{\infty}$ given by the Fully-Corrective Block-Coordinate Frank-Wolfe (Algorithm 1) with sampling approximation (24), (25) has*

$$G(\boldsymbol{\alpha}^t) - G^* \leq \frac{4(QR^2 + \Delta G^0)}{t/N\nu + 2} \ , \ t \geq 0 \tag{28}$$

for $t \leq (2QR^2\nu)/\epsilon_d$.

4. Practical Issues

4.1. Efficient Implementation for Sparse Data Matrix

When $nnz(\boldsymbol{x}_i) \ll D$, it is crucial to exploit sparsity of both $\boldsymbol{w}_k$ and $\boldsymbol{x}_i$ in the computation of search step (20), (24) or (25). Let $\boldsymbol{z}_i$ be a length-K vector with $z_{ik} = \langle \boldsymbol{w}_k, \boldsymbol{x}_i \rangle$. We maintain $\{\boldsymbol{w}^j\}_{j=1}^{D}$, where $\boldsymbol{w}^j := (w_{jk})_{k \in [K]}$, as sparse vectors using hashtable implementation specified in Sec. 4.2, and compute $\boldsymbol{z}_i$ via $\boldsymbol{z}_i = \sum_{j:x_{ij} \neq 0} x_{ij} \boldsymbol{w}_j$ which can be realized in time $O(nnz(\boldsymbol{x}_i) nnz(\boldsymbol{w}_j))$. And when sampling (25) is used, we divide $\boldsymbol{w}_j$ into ν sparse vectors $\{\boldsymbol{w}_j^q\}_{q=1}^{\nu}$ maintained separately, and compute (25) via

$$\boldsymbol{z}_i^q = \sum_{j \in \mathcal{D}_i} x_{ij} \boldsymbol{w}_j^q \tag{29}$$

where $\boldsymbol{z}_i^q$ is of length K/ν.

4.2. Hashing under Limited Memory

In Extreme Classification setting, both the $N \times K$ matrix A and $D \times K$ matrices W, V cannot be stored as array in memory. On the other hand, a general-purpose hash table is significantly slower than array due to expensive hash function evaluation. In our implementation, each matrix is stored as an array of hashtables that share the same hash function $h : [K] \to [K]$ constructed by a random permutation of $[K]$, so the hash function only needs to be evaluated once for all hashtables, giving an efficiency close to array.

4.3. Post Processing after Variable Selection

The purpose of using ℓ_1 regularization is to select a small subset of primal variables for scalability. However, in some cases, optimizing a purely ℓ_2-regularized problem

Table 1. Results on multiclass data sets: N= number of train samples, K = number of classes, D = number of features, the best results among all solvers are marked. Multi-SVM is not applicable to Dmoz due to $> 200G$ memory requirement. We mark accuracy of PD-Sparse with "*" if the post processing discussed in Sec. 4.3 improves accuracy.

Data		**FastXML**	**LEML**	**1vsA-Logi**	**1vsA-SVM**	**Multi-SVM**	**1vsA-L1-Logi**	**PD-Sparse**	**VW(oaa)**	**VW(Tree)**	**SLEEC**
LSHTC1	train time	2131s	78950s	$\approx$ 6d	23744s	6411s	$\approx$ 14d	**952s**	28129s	1193s	10793s
N=83805	model size	308M	7.7G	$\approx$ 57G	11G	4.5G	$\approx$ **57M**	92M	2.4G	744M	1.38G
D=347255	predict time	6.33s	189s	N/A	50.3s	49.0s	N/A	**6.20s**	80.0s	6.84s	155s
K=12294	test acc(%)	21.66	16.52	N/A	**23.22**	22.4	N/A	22.66^*	14.4	10.56	12.8
Dmoz	train time	6900s	97331s	$\approx$ 27d	136545s	N/A	$\approx$ 565d	**2068.14s**	361943s	7103s	113200s
N=345068	model size	1.5G	3.6G	$\approx$ 96G	19G	N/A	$\approx$ 406M	**40M**	4.3G	1.8G	3.23G
D=833484	predict time	57.1s	1298s	N/A	429.7s	N/A	N/A	**6.74s**	548.3s	28s	3292s
K=11947	test acc(%)	38.4	31.28	N/A	36.8	N/A	N/A	**39.58**	35.44	21.27	32.49
imgNet	train time	28440s	107490s	380971s	28640s	14510s	472611s	**4958s**	9283s	6492s	520570s
N=1261404	model size	914M	13M	14M	23M	24M	**1.8M**	3.6M	7.7M	35M	2.76G
D=1000	predict time	139s	554s	315.3s	136.8s	203.4s	390.5s	329.5s	191s	**37.7s**	45372s
K=1000	test acc(%)	6.48	7.21	8.56	**15.25**	10.3	10.07	12.7	8.51	5.37	8.5
aloi.bin	train time	2410s	62440s	42390s	9468s	449s	31770s	773.8s	2097.4s	**334.3s**	12200s
N=100000	model size	992M	5.4G	15G	5.9G	612M	18M	**7.1M**	1.9G	106M	1.96G
D=636911	predict time	10.99s	38.83s	16.61s	22.83s	12.42s	13.24s	**1.42s**	14.01s	1.59s	191s
K=1000	test acc(%)	95.5	88.16	96.34	96.63	**96.66**	95.71	96.33	96.54	89.47	92.55
sector	train time	100.77s	556.31s	107.12s	19.46s	**11.46s**	102.31s	14.12s	153.7s	327.34s	164.3s
N=8658	model size	7.0M	48M	129M	62M	57M	**580K**	1.6M	47M	17M	223.5M
D=55197	predict time	0.25s	**0.069s**	0.114s	0.156s	0.169s	0.13s	0.09s	0.28s	0.16s	1.59s
K=105	test acc(%)	84.9	94.07	90.8	94.79	95.11	93.13	$\mathbf{95.3}^*$	92.09	82.1	87.62

with features selected by ℓ_1 regularization yields better performance. In our solver, we implement a post-processing step which solves a ℓ_2-regularized problem with only primal and dual variables in the active sets obtained by solving (13). We construct a $N_k \times D_k$ data submatrix for each label which stores an element x_{ij} only if $\alpha^*_{ik} \neq 0$ "and" $w^*_{jk} \neq 0$ for the optimal solution $(\boldsymbol{\alpha}^*, \boldsymbol{w}^*)$ of (13). Note the data submatrix of each label is extremely small given the sparsity of $\boldsymbol{\alpha}^*$, $\boldsymbol{w}^*$. Therefore, one can solve the post-processing ℓ_2-regularized problem very efficiently.

4.4. Parameters for Bi-stochastic Search

In practice, we set

$$R_w^2 = \frac{CNd_i}{DK} \gtrsim \frac{d_i}{DK}\sum_{k=1}^{K}\|\boldsymbol{w}_k^t\|_2^2 \approx \sup_{k,i}\{\sum_{j:\boldsymbol{x}_{ij}\neq 0}(\boldsymbol{w}_{kj}^t)^2\},$$

where the first inequality is from (40) (Appendix C), and the second approximation is an assumption on distribution of W. Suppose $\boldsymbol{x}_i$ is scaled so $\|\boldsymbol{x}_i\|_\infty \leq 1$. (26) becomes

$$\tilde{d}_i \gtrsim d_i \frac{CNd_i \log(\frac{K}{\delta})}{\epsilon_d^2 DK} \qquad (30)$$

We use $nnz(X)$ to esimate Nd_i and set the speed up rate to be $\frac{d_i}{\tilde{d}_i} = \lceil \min(\frac{5DK}{Cnnz(X)\log K}, \frac{nnz(X)}{5N})\rceil$, $\forall i$ in our implementation. To further amortize the cost of greedy search, for each iteration, if the amount of time spent on Greedy Search is more than twice of the time spent on the rest of other operations, we double the number of coordinates (with largest scores) returned by the search (25).

5. Experiments

In this section we compare our proposed Primal-Dual Sparse(PD-Sparse) method with existing approaches to multiclass and multilabel problems. In all experiments, we set C=1 for all methods based on Empirical Risk Minimization, and choose $\nu = 3$ and $\lambda \in \{0.01, 0.1, 1, 10\}$ that gives best accuracy on a heldout data set for our method. To prevent over-fitting, we compute accuracy on a heldout data set to determine number of iterations used in all the iterative solvers. The compared algorithms are listed as follows.

- LibLinear (Fan et al., 2008) one-versus-all logistic regression (1vsA-Logi).
- LibLinear one-vs-all SVM (1vsA-SVM).
- LibLinear multiclass SVM (Multi-SVM).
- LibLinear one-vs-all l_1-regularized logistic regression solver (1vsA-L1-Logi).
- Vowpal-Wabbit (VW): A public fast learning system proposed in (Choromanska & Langford, 2015) for Extreme multiclass classification. We use one-against-all (oaa) and online trees (Tree) options provided by their solver.
- FastXML: An Extreme multilabel classification method (Prabhu & Varma, 2014) that organizes models with tree structure. We use solver provided by the author with default parameters.
- LEML: A low-rank Empirical-Risk-Minimization solver from (Yu et al., 2013). We use solver provided by the authors with rank chosen to be lowest

Table 2. Results on Multilabel data sets, N= number of training samples, K = number of classes, D = number of features. The accuracy means ”top-1” accuracy (precision). Note that SLEEC's performance highly depends on a set of 9 parameters. Here results are obtained using default parameters of the solver (given in Table 4, Appendix B). An optimized parameter set may give better result.

Data		**FastXML**	**LEML**	**1vsA-Logi**	**1vsA-SVM**	**1vsA-L1-Logi**	**PD-Sparse**	**SLEEC**
LSHTC-wiki	train time	**104442s**	217190s	$>$10y	$>$96d	$>$10y	124867s	2224000s
N=2355436	model size	8.9G	10.4G	$\approx$ 426G	$\approx$870G	$\approx$ **358M**	685M	12.6G
D=2085167	predict time	164.8s	2896s	N/A	N/A	N/A	**15.56s**	8906s
K=320338	test acc %	78.28	28.46	N/A	N/A	N/A	$\mathbf{89.3}^*$	73.44
EUR-Lex	train time	**317s**	7471s	22551s	3227s	32531s	434.9s	2443s
N=15643	model size	324.5M	78M	257M	118M	14M	**8.0M**	80.8M
D=5000	predict time	**0.996s**	42.24s	7.93s	7.23s	1.39s	1.089s	4.89s
K=3956	test acc %	67.3	67.82	**77.3**	64.5	73.8	76.3	74.2
RCV1-regions	train time	94.06s	2247s	79.27s	14.73s	84.74s	**8.82s**	1129s
N=20835	model size	14.61M	205M	129M	39M	**504K**	1.7M	204M
D=47237	predict time	0.824s	2.515s	0.486s	0.392s	0.174s	**0.115s**	15.8s
K=225	test acc %	93.28	96.28	90.96	95.98	94.7	**96.54**	91
bibtex	train time	18.35s	157.9s	8.944s	**3.24s**	13.97s	5.044s	298s
N=5991	model size	27M	8.6M	3.7M	3.3M	412K	**68K**	26.7M
D=1837	predict time	0.09s	0.2215s	0.0383s	0.079s	0.0238s	**0.0059s**	0.94s
K=159	test acc %	64.14	64.01	62.65	58.46	61.16	64.55	**65.09**

from $\{50, 100, 250, 500, 1000\}$ that gives accuracy on heldout data set not worse than the best by 1%.

- SLEEC: A method based on Sparse Local Embeddings for Extreme multilabel classification (Bhatia et al., 2015). We use solver provided by the author with default parameters.

Among these solvers, LibLinear multiclass SVM, VowpalWabbit are only for multiclass problems. All other solvers can be used on both multiclass and multilabel data sets. Note FastXML, LEML and SLEEC are designed for multilabel problems but also applicable to multiclass problems.

Our experiments are conducted on 9 public data sets. Among them, *LSHTC1*, *Dmoz*, *imagenet*, *aloi.bin* and *sector* are multiclass and *LSHTC-wiki*, *EUR-Lex*, *RCV1-regions*, *bibtex* are multilabel. *ImageNet* uses bag-of-word features downloaded directly from ImageNet [2]. *EUR-Lex* and *bibtex* are from Mulan multilabel data collections. [3] *LSHTC1*, *Dmoz* and *LSHTC-wiki* are from LSHTC2 competition described in (Partalas et al., 2015). *RCV1-regions*, *aloi.bin* and *sector* are from LIBSVM data collection [4], where *aloi.bin* uses Random Binning features (Rahimi & Recht, 2007; Yen et al., 2014) approximating effect of RBF Laplacian kernel.

The statistics of data sets and results are shown in Table 1 and 2. We include statistics of test and heldout data set in Appendix B. Note many one-vs-all solvers require running for a huge amount of time. We run a distributed version and use training time and models of at least 100 classes to estimate the expected total running time and model size.

Table 3. Average number of active dual and primal variables (k_A, k_W respectively) when parameter λ maximizes heldout accuracy.

Data sets	k_A	k_W
EUR-Lex (K=3956)	20.73	45.24
LSHTC-wiki (K=320338)	18.24	20.95
LSHTC (K=12294)	7.15	4.88
aloi.bin (K=1000)	3.24	0.31
bibtex (K=159)	18.17	1.94
Dmoz (K=11947)	5.87	0.116

As showed in the table, solvers rely on structural assumptions such as FastXML (tree), VW (tree), LEML (low-rank) and SLEEC (piecewise-low-rank) could obtain accuracy significantly worse than standard one-vs-all methods on multiclass data sets. Standard multiclass solvers however suffer from complexity growing linearly with K. On the other hand, by exploiting primal and dual sparsity inherent in Extreme Classification problem, PD-Sparse has training time, prediction time and model size growing sublinearly with K while keeping a competitive accuracy. As showed in Table 3, the average number of active dual variables for each sample is much smaller than the number of classes.

Acknowledgements

This research was supported by NSF grants CCF-1320746 and IIS-1546459, IIS-1149803, IIS-1320894, IIS-1447574, and DMS-1264033, ARO grants W911NF-12-1-0390, and NIH grant R01 GM117594-01 as part of the Joint DMS/NIGMS Initiative to Support Research at the Interface of the Biological and Mathematical Sciences.

[2] `http://image-net.org/`

[3] `mulan.sourceforge.net/datasets-mlc.html`

[4] `www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html`

References

Bhatia, Kush, Jain, Himanshu, Kar, Purushottam, Varma, Manik, and Jain, Prateek. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pp. 730–738, 2015.

Chen, Yao-Nan and Lin, Hsuan-Tien. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems*, pp. 1529–1537, 2012.

Choromanska, Anna, Agarwal, Alekh, and Langford, John. Extreme multi class classification. In *NIPS Workshop: eXtreme Classification, submitted*, 2013.

Choromanska, Anna E and Langford, John. Logarithmic time online multiclass prediction. In *Advances in Neural Information Processing Systems*, pp. 55–63, 2015.

Crammer, Koby and Singer, Yoram. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2: 265–292, 2002.

Crammer, Koby and Singer, Yoram. A family of additive online algorithms for category ranking. *The Journal of Machine Learning Research*, 3:1025–1058, 2003.

Deng, Jia, Berg, Alexander C, Li, Kai, and Fei-Fei, Li. What does classifying more than 10,000 image categories tell us? In *Computer Vision–ECCV 2010*, pp. 71–84. Springer, 2010.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

Kakade, Sham, Shalev-Shwartz, Shai, and Tewari, Ambuj. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, 2009.

Kapoor, Ashish, Viswanathan, Raajay, and Jain, Prateek. Multilabel classification using bayesian compressed sensing. In *Advances in Neural Information Processing Systems*, pp. 2645–2653, 2012.

Keerthi, S Sathiya, Sundararajan, Sellamanickam, Chang, Kai-Wei, Hsieh, Cho-Jui, and Lin, Chih-Jen. A sequential dual method for large scale multi-class linear svms. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 408–416. ACM, 2008.

Lacoste-Julien, Simon, Jaggi, Martin, Schmidt, Mark, and Pletscher, Patrick. Block-coordinate frank-wolfe optimization for structural svms. In *ICML 2013 International Conference on Machine Learning*, pp. 53–61, 2013.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Meshi, Ofer, Mahdavi, Mehrdad, and Schwing, Alex. Smooth and strong: Map inference with linear convergence. In *Advances in Neural Information Processing Systems*, pp. 298–306, 2015.

Partalas, Ioannis, Kosmopoulos, Aris, Baskiotis, Nicolas, Artieres, Thierry, Paliouras, George, Gaussier, Eric, Androutsopoulos, Ion, Amini, Massih-Reza, and Galinari, Patrick. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*, 2015.

Prabhu, Yashoteja and Varma, Manik. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 263–272. ACM, 2014.

Rahimi, Ali and Recht, Benjamin. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2007.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Tibshirani, Ryan J et al. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490, 2013.

Yen, Ian En-Hsu, Lin, Ting-Wei, Lin, Shou-De, Ravikumar, Pradeep K, and Dhillon, Inderjit S. Sparse random feature algorithm as coordinate descent in hilbert space. In *Advances in Neural Information Processing Systems*, pp. 2456–2464, 2014.

Yu, Hsiang-Fu, Jain, Prateek, Kar, Purushottam, and Dhillon, Inderjit S. Large-scale multi-label learning with missing labels. *arXiv preprint arXiv:1307.5101*, 2013.

Zou, Hui and Hastie, Trevor. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320, 2005.

6. Appendix A: Convergence Proof

The proofs of Theorem 1, 2 are similar to that in (Lacoste-Julien et al., 2013). To be self-contained, we provide proofs in the following.

6.1. Proof for Theorem 1

The dual problem (14) has (generalized) Hessian for i-th block of variable $\boldsymbol{\alpha}^i$ being upper bounded by

$$\nabla^2_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}) \preceq Q_i I.$$

where $Q_i = \|\boldsymbol{x}_i\|^2$. Since the active set includes the most-violating pair (19) that defines the Frank-Wolfe direction $\boldsymbol{\alpha}^t_{FW}$ satisfying (18), the update given by solving the active-set subproblem (21) has

$$\begin{aligned} & G(\boldsymbol{\alpha}^{t+1}) - G(\boldsymbol{\alpha}^t) \\ & \leq \gamma \langle \nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^{it}_{FW} - \boldsymbol{\alpha}^{it} \rangle + \frac{Q_i \gamma^2}{2} \|\boldsymbol{\alpha}^{it}_{FW} - \boldsymbol{\alpha}^{it}\|^2 \\ & \leq \gamma \langle \nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^{it}_{FW} - \boldsymbol{\alpha}^{it} \rangle + \frac{Q_i R^2 \gamma^2}{2} \end{aligned}$$

for any $\gamma \in [0, 1]$, where $\|\boldsymbol{\alpha}^t_{FW} - \boldsymbol{\alpha}^{it}\|^2 \leq R^2 = 4C^2$ since both $\boldsymbol{\alpha}^t_{FW}$, $\boldsymbol{\alpha}^{it}$ lie within the domain (16). Taking expectation w.r.t. i (uniformly sampled from $[N]$), we have

$$\begin{aligned} & E[G(\boldsymbol{\alpha}^{t+1})] - G(\boldsymbol{\alpha}^t) \\ & \leq \frac{\gamma}{N} \langle \nabla_{\boldsymbol{\alpha}} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^t_{FW} - \boldsymbol{\alpha}^t \rangle + \frac{QR^2\gamma^2}{2N} \end{aligned} \tag{31}$$

where $Q = \sum_{i=1}^N Q_i$. Then denote $\boldsymbol{\alpha}^*$ as an optimal solution, by convexity and the definition of Frank-Wolfe direction we have

$$\begin{aligned} \langle \nabla_{\boldsymbol{\alpha}} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^t_{FW} - \boldsymbol{\alpha}^t \rangle & \leq \langle \nabla_{\boldsymbol{\alpha}} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^* - \boldsymbol{\alpha}^t \rangle \\ & \leq G^* - G(\boldsymbol{\alpha}^t), \end{aligned}$$

where $G^* := G(\boldsymbol{\alpha}^*)$. Together with (31), we have

$$\Delta G^{t+1} - \Delta G^t \leq \frac{-\gamma}{N} \Delta G^t + \frac{QR^2\gamma^2}{2N} \tag{32}$$

for any $\gamma \in [0, 1]$, where $\Delta G^t := E[G(\boldsymbol{\alpha}^t)] - G^*$. By choosing $\gamma = \frac{2N}{t+2N}$, the recurrence (32) leads to the result

$$\Delta G^t \leq \frac{2(QR^2 + \Delta G^0)}{t/N + 2},$$

which can be verified via induction as in the proof of Lemma C.2 of (Lacoste-Julien et al., 2013).

6.2. Proof for Theorem 2

The approximation criteria (24) searches active label from one out of ν partitions of $[K]$. Suppose in the t-th iteration, a subset not containing most-violating label (20) was chosen, we have

$$G(\boldsymbol{\alpha}^{t+1}) - G(\boldsymbol{\alpha}^t) \leq 0 \tag{33}$$

and suppose a subset containing most-violating label was chosen, we have

$$\begin{aligned} & G(\boldsymbol{\alpha}^{t+1}) - G(\boldsymbol{\alpha}^t) \\ & \leq \gamma \langle \nabla_{\boldsymbol{\alpha}^i} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^{it}_{FW} - \boldsymbol{\alpha}^{it} \rangle + \frac{Q_i R^2 \gamma^2}{2} + \gamma \epsilon_d \end{aligned} \tag{34}$$

where ϵ_d is the error caused by sampling (25). Since (33), (34) happen with probabilities $1-1/\nu$ and $1/\nu$ respectively, we have expected descent amount

$$\begin{aligned} & E[G(\boldsymbol{\alpha}^{t+1}) - G^*] - (G(\boldsymbol{\alpha}^t) - G^*) \\ & \leq \frac{\gamma}{N\nu} \langle \nabla_{\boldsymbol{\alpha}} G(\boldsymbol{\alpha}^t), \boldsymbol{\alpha}^t_{FW} - \boldsymbol{\alpha}^t \rangle + \frac{QR^2\gamma^2}{2N\nu} + \frac{\gamma \epsilon_d}{\nu} \\ & \leq \frac{-\gamma}{N\nu} (G(\boldsymbol{\alpha}^t) - G^*) + \frac{QR^2\gamma^2}{2N\nu} + \frac{\gamma \epsilon_d}{\nu}. \end{aligned} \tag{35}$$

following the same reasoning of (31) and (32). For

$$\epsilon_d \leq \frac{QR^2\gamma}{2N},$$

we have

$$\begin{aligned} & E[G(\boldsymbol{\alpha}^{t+1}) - G^*] - (G(\boldsymbol{\alpha}^t) - G^*) \\ & \leq \frac{-\gamma}{N\nu} (G(\boldsymbol{\alpha}^t) - G^*) + \frac{QR^2\gamma^2}{N\nu}. \end{aligned} \tag{36}$$

Therefore, by choosing $\gamma = \frac{2}{t/(N\nu)+2}$, we have

$$\Delta G^t \leq \frac{4(QR^2 + \Delta G^0)}{t/(N\nu) + 2}$$

for t satisfying

$$0 \leq t \leq \nu QR^2 / \epsilon_d.$$

[4] http://research.microsoft.com/en-us/um/people/manik/code/SLEEC/download.html

7. Appendix B: Additional Statistics

Table 4. Default parameter setting used in SLEEC's code. One might need to refer to their webpage [6]for explanation of parameters.

num_learners	**num_clusters**	**SVP_neigh**
5	5	50
out_Dim	**w_thresh**	**sp_thresh**
75	0.75	0.5
cost	**NNtest**	**normalize**
0.1	20	1

Table 5. Statistics for heldout and test data set

Data Sets	**Train Size**	**Heldout Size**	**Test Size.**
LSHTC-wiki	2355436	5000	5000
EUR-Lex	15643	1738	1933
bibtex	5991	665	739
RCV1-regions	20835	2314	5000
LSHTC	83805	5000	5000
aloi.bin	90000	10000	8000
Dmoz	310562	34506	38340
ImageNet	1125264	10000	126140
sector	7793	865	961

8. Appendix C: Bounds for Approximation (25)

Let σ_{ki}^2 be the variance of $\bar{C}_k(\mathcal{D}_i)$. We have

$$\sigma_{ki}^2 \leq \hat{\sigma}_{ki}^2 = \frac{1}{\tilde{d}_i}\|\boldsymbol{x}_i\|_1\|\boldsymbol{x}_i\|_\infty R_w^2 \leq \frac{d_i}{\tilde{d}_i}\|\boldsymbol{x}_i\|_\infty^2 R_w^2 \tag{37}$$

, where R_w^2 is an upper bound on $\sum_{j:\boldsymbol{x}_{ij}\neq 0}(\boldsymbol{w}_{kj}^t)^2$.

For $\epsilon = O(\|\boldsymbol{x}_i\|_1 R_w)$, Bernstein-Type inequality gives

$$\Pr[|\bar{C}_k(\mathcal{D}_i) - \langle \boldsymbol{w}_k^t, \boldsymbol{x}_i\rangle| > \epsilon] \leq e^{-\frac{\epsilon^2}{2\hat{\sigma}_{ki}^2}} \tag{38}$$

Suppose we want to approximate $\langle \boldsymbol{w}_k^t, \boldsymbol{x}_i\rangle$ within ϵ_d for all $k \in [K]$ with failure probability at most δ. Combining (37), (38) and using union bound, we only need

$$\frac{d_i}{\tilde{d}_i} \lesssim \frac{\epsilon_d^2}{\log(\frac{K}{\delta})\|\boldsymbol{x}_i\|_\infty^2 R_w^2} \tag{39}$$

Also, look at the dual objective function in (14), initially we have $G(\boldsymbol{\alpha}) = G(\mathbf{0}) = 0$. Since our method is dual-descent, we have $G(\boldsymbol{\alpha}^t) \leq 0$, thus

$$\frac{1}{2}\sum_{k=1}^{K}\|\boldsymbol{w}_k^t\|_2^2 \leq -\sum_{i=1}^{N}\boldsymbol{e}_i^T\boldsymbol{\alpha}^i \leq CN \tag{40}$$

where the last inequality follows from (16).

[6] http://research.microsoft.com/en-us/um/people/manik/code/SLEEC/download.html