New Optimization Methods for Large Scale SVMs

Yifan Kang YIFANK@CLEMSON.EDU
Yarui Cao YARUIC@CLEMSON.EDU
Kai Liu KAIL@CLEMSON.EDU
Clemson University, USA

Abstract

We revisit the classical support vector machine (SVM) from a duality perspective and propose new optimization methods that yield both theoretical and practical advantages. For binary L2 SVM, we utilize proximal gradient method where in each iteration we derive a novel dual formulation to deal with equation constraint. The Lagrangian multiplier can be obtained fast due to its monotone property. For the multiclass setting, we introduce a scalable optimization algorithm based on the Frank-Wolfe method applied to a structured dual of the multiclass SVM problem. Unlike reformulating into a classical quadratic programming (QP), our method is scalable to problems with many samples and classes. Empirical experiments demonstrate that our methods is significantly superior in terms of way less time consumption on large-scale datasets.

1. Introduction

Support Vector Machines (SVMs) have long been a cornerstone of supervised learning, offering strong theoretical guarantees and competitive empirical performance across a wide range of classification and regression tasks. Introduced in the 1990s, the classical SVM formulation casts the learning problem as a convex optimization task, enabling global solutions and robust generalization [13]. Despite the rapid adoption of deep neural networks in recent years, SVMs remain relevant, particularly in settings where interpretability, theoretical robustness, and computational efficiency are prioritized [1, 11]. Traditional SVM formulations (L1-SVM) have been extensively studied, with efficient solvers developed for both binary and multiclass classification. However, two important challenges persist. First, L2-SVM training to massive datasets with millions of examples and features remains nontrivial. Second, extending the SVM framework to handle structured outputs or extremely high-dimensional multiclass settings introduces computational bottlenecks that are not well-addressed by existing methods given QP is considerably time consuming. To address these limitations, recent research has revisited SVM optimization through *Frank-Wolfe algorithms* [9], and *dual decomposition techniques* [10]. These approaches offer new opportunities to rethink the classical SVM, particularly for modern applications where scalability and adaptivity are critical.

In this paper, we propose new algorithmic discoveries in the SVM framework that (i) leverage proximal gradient method based on the dual formulation of L2-SVM problem, where in each step of proximal gradient, a (second) dual method is utilized and (ii) develop a Frank-Wolfe-based solver tailored to large-scale multiclass SVMs. Our methods are motivated by the need to handle datasets with millions of examples/features and thousands of classes, where standard solvers (such as QP) become prohibitively expensive or inpractical. The contributions of this work are as follows:

- We present a novel optimization method based on the dual of L2-SVM that allows for faster convergence, which is a total white-box, first order optimization method.
- We develop a scalable Frank-Wolfe optimization algorithm for multiclass SVMs, providing convergence guarantees and complexity analysis.
- We conduct experiments on large-scale benchmark datasets, demonstrating significant improvements in efficiency compared to existing solvers.

2. L1-SVM: Primal and Dual Formulations

Let $(x_i, y_i)_{i=1}^n$ be a binary classification dataset with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. Classical (L1-) SVM aims to find a linear classifier that maximizes the margin while penalizing hinge loss errors.

Primal Problem The primal optimization problem for L1-SVM is:

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \xi_i, \text{ s.t. } y_i(w^\top x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, n. \tag{1}$$

Here, $w \in \mathbb{R}^d$ is the weight vector, $b \in \mathbb{R}$ is the bias term, ξ_i are slack variables, and $C > 0$ is a regularization parameter controlling the trade-off between margin size and hinge loss penalty.

Dual Problem To derive the dual, we introduce Lagrange multipliers $\alpha_i \geq 0$ for the margin constraints, and $\mu_i \geq 0$ for the non-negativity constraints. Eliminating primal variables via KKT conditions yields the dual:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j x_i^\top x_j, \text{ s.t. } 0 \leq \alpha_i \leq C,\ \sum_{i=1}^n \alpha_i y_i = 0. \tag{2}$$

To obtain $\boldsymbol{\alpha}^*$, one can make use of Sequential Minimal Optimization (SMO) [13], Quadratic Programming (QP) [16] based on Interior-Point Method (IPM) or Frank-Wolfe method [7] given the compactness of the domain. To predict new/testing data x, the decision function is given by: $\text{sign}(b + \sum_{i=1}^n \alpha_i y_i x_i^\top x)$. One can see that for primal formulation in Eq. (1), QP can also be applied. However, for kernel (such as Gaussian) version, solving Eq. (2) is the only option.

3. L2-SVM: A Dual of Dual Approach

Similar to the last section, one can formulate the objective as:

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{2}\sum_{i=1}^n \xi_i^2, \text{ s.t. } y_i(w^\top x_i + b) \geq 1 - \xi_i,\ i = 1, \ldots, n. \tag{3}$$

Different from L1-SVM, we remove the nonnegative constraint on ξ_i since $\xi_i = 0$ will always yiled lower objective than when ξ_i is negative, while satisfying the inequality constraint automatically. By introducing Lagrangian multipliers, we have the dual problem:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j x_i^\top x_j - \frac{1}{2C}\sum_{i=1}^n \alpha_i^2, \text{ s.t. } 0 \leq \alpha_i,\ \sum_{i=1}^n \alpha_i y_i = 0. \tag{4}$$

The above can still be cast into a standard QP formulation, however, when the sample size (n) is huge, it can be time-consuming as the complexity of QP typically is $\mathcal{O}(n^3)$. To our best knowledge, our proposed algorithm to be discussed is the first try to solve the dual from its dual.

Obviously, Eq. (4) is equivalent to

$$\min_{\boldsymbol{\alpha}} \quad f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\top \mathcal{K}\boldsymbol{\alpha} + \frac{1}{2C}\|\boldsymbol{\alpha}\|_2^2 - \boldsymbol{\alpha}^\top \mathbf{1}, \text{ s.t. } 0 \leq \boldsymbol{\alpha}, \ \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle = 0, \tag{5}$$

where $\mathcal{K}(i,j) = y_i y_j x_i^\top x_j$ and matrix $\mathcal{K}$ is Symmetric Positive Semi-Definite (*SPSD*). Given the (strong) convexity of the objective, we can make use of proximal gradient descent where the simplest option of stepsize is short step $1/L := 1/[\sigma_1(\mathcal{K}) + 1/C]$ where $\sigma_1(\cdot)$ denotes the largest singular value. It is worth noting that any backtrackling line search option is also permitted. Therefore, in each iteration it boils down to

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_2^2, \text{ s.t. } 0 \leq \boldsymbol{\alpha}, \ \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle = 0, \tag{6}$$

where $\boldsymbol{\beta} = \boldsymbol{\alpha}_{old} - \frac{1}{L}\nabla f(\boldsymbol{\alpha}_{old})$. Now let λ be the Lagrange multiplier to the linear equality constraint $\langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle = 0$ and the Lagrangian function therefore is

$$L(\boldsymbol{\alpha}, \lambda) = \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_2^2 + 2\lambda\langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle = \|\boldsymbol{\alpha} - \boldsymbol{\beta} + \lambda \boldsymbol{y}\|_2^2 + \text{terms irrevalent to } \boldsymbol{\alpha}. \tag{7}$$

Apparently $\boldsymbol{\alpha} = [\boldsymbol{\beta} - \lambda \boldsymbol{y}]_+$ given the nonnegative constraint. By plugging it back to the equality constraint $\langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle = 0$, we have $\sum_{i=1}^n y_i[\beta_i - \lambda y_i]_+ = 0$. Apparently the left side in the above equation is monotonically decreasing with respect to λ (for both $y_i = 1$ and $y_i = -1$ cases), where the root can be found for example by a simple bisection procedure [2]. We summarize our method in the following algorithm. One can easily tell that the proposed method is scalable to large data as each step is computationally economic, where there is no *inversion* or *svd* operation.

Algorithm 1: Proposed method to solve Eq. (5) for L2-SVM

```
Data: Given $C$, $(x_i, y_i)_{i=1}^n$, formulate $\mathcal{K}$, calculate $L = \sigma_1(\mathcal{K}) + 1/C$ and initialize $\boldsymbol{\alpha} = \mathbf{0}$
Result: Obtain optimal $\boldsymbol{\alpha}$ to Eq. (5)
while not converge do
    calculate $\boldsymbol{\beta} = \boldsymbol{\alpha} - \frac{1}{L}(\mathcal{K}\boldsymbol{\alpha} + \frac{\boldsymbol{\alpha}}{C} - \mathbf{1})$;
    find $\lambda$ to the equation $\sum_{i=1}^n y_i[\beta_i - \lambda y_i]_+ = 0$ via bisection procedure;
    obtain $\boldsymbol{\alpha} = [\boldsymbol{\beta} - \lambda \boldsymbol{y}]_+$;
end
```

We conclude this section by pointing out that Nesterov Acceleration Gradient can be applied to obtain a convergence rate of $\mathcal{O}(1/t^2)$ [4] (t is the counter for *while*-loop iterations).

4. Multiclass SVM — Matrix-wise Frank-Wolfe Approach

Binary classification is the foundation for multi-classification, since theoretically one can always use One-vs-One or One-vs-All to that end. However, One-vs-All might fail to find a good predictor [14] while One-vs-One needs to train $K(K-1)/2$ hyperplanes, which is prohibitive when there are many classes. Therefore, researchers aim to find more direct methods and among them, Crammer–Singer formulation is one of the most widely used ones.

Crammer–Singer Multiclass SVM Given training data $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$, $y_i \in \{1, \ldots, K\}$. Let $W = [w_1, \ldots, w_K] \in \mathbb{R}^{d\times K}$ collect class weight vectors. Crammer–Singer (CS) primal is [6]

$$\min_{W,\xi} \quad \frac{1}{2}\sum_{k=1}^{K}\|w_k\|^2 + C\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } \forall i,\ \forall k \neq y_i:\ w_{y_i}^\top \phi(x_i) - w_k^\top \phi(x_i) \ \geq\ 1-\xi_i,\ \xi_i \geq 0. \tag{8}$$

The decision function learned is of the form $x \to \arg\max_l \langle w_l, \phi(x)\rangle$. Defining the Lagrangian and applying KKT conditions leads to the following equivalent dual optimization problem, which can be expressed in terms of the kernel function $\mathcal{K}$ where $\mathcal{K}(i,j) = \langle \phi(x_i), \phi(x_j)\rangle$:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{n\times K}} f(\boldsymbol{\alpha}) = \frac{1}{2}\operatorname{trace}(\boldsymbol{\alpha}^\top \mathcal{K}\boldsymbol{\alpha}) - \langle \boldsymbol{\alpha}, \mathbf{1}_{ind}\rangle,\ \text{s.t. } \forall i: (0 \leq \boldsymbol{\alpha}_{iy_i} \leq C) \wedge (\forall j \neq y_i, \boldsymbol{\alpha}_{ij} \leq 0) \wedge (\boldsymbol{\alpha}\mathbf{1} = \mathbf{0}), \tag{9}$$

where $\mathbf{1}_{ind} \in \mathbb{R}^{n\times K}$ is the indicator matrix denoting each sample belonging to the corresponding class. The inference for x_{test} is given by $[\sim, ind] = \max[\phi^\top(x_{test})\Phi(x_{train})\boldsymbol{\alpha}]$. By observing the quadratic objective and its feasible domain, we believe Frank-Wolfe (FW) is the best option for optimization. We notice previous work [9] has utilized FW for structured SVMs, however, we stipulate the significant difference: Lacoste-Julien et al. [9] studies a broader class problem — structured SVMs, while we only focus on multiclass SVM. Their conclusion holds more generally while for our specific topic (multi-class SVMs), we make further improvements. They update each block stochastically with FW, while we update as a whole matrix. Theoretically, our proposed algorithm can be way faster than the counterpart which will be quantilized in later section. Second, we use a closed-form step rather than 'oblivious' stepsize $2n/(t+2n)$, whereas the optimal stepsize for each block results in trivial difference as the update are almost the same. See Section 5.

Algorithm 2: Proposed method to solve Eq. (9) for multiclassification SVM

```
Data: Given $C$, $(x_i, y_i)_{i=1}^n$, formulate $\mathcal{K}$ and $\mathbf{1}_{ind}$, and initialize $t = 0, \boldsymbol{\alpha} = \mathbf{0}$
Result: Obtain optimal $\boldsymbol{\alpha}$ to Eq. (9)
while $t \leq T$ do
    $\boldsymbol{s}_t \in \arg\min_{\boldsymbol{s}\in\mathcal{C}}\langle \nabla f(\boldsymbol{\alpha}_t), \boldsymbol{s}\rangle = \arg\min_{\boldsymbol{s}\in\mathcal{C}}\langle \mathcal{K}\boldsymbol{\alpha}_t - \mathbf{1}_{ind}, \boldsymbol{s}\rangle$;
    $\boldsymbol{d}_t = \boldsymbol{s}_t - \boldsymbol{\alpha}_t$;
    set step size by line search $\gamma_t = \arg\min_{\gamma\in[0,1]} f(\boldsymbol{\alpha}_t + \gamma \boldsymbol{d}_t)$;
    $\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \gamma_t \boldsymbol{d}_t$, $t = t+1$;
end
```

The first line in the *while* loop is to determine $\boldsymbol{s}_t$, which can be optimized row by row given the constraint in Eq. (9). Specifically, denote the i-th row of $\nabla f(\boldsymbol{\alpha}_t)$ as $\boldsymbol{q}$: if $\boldsymbol{q}(y_i) == \max(\boldsymbol{q})$, then i-th row of $\boldsymbol{s}_t$ is $\mathbf{0}$; otherwise $\boldsymbol{s}_t(i, y_i) = C, \boldsymbol{s}_t(i, \arg\max \boldsymbol{q}) = -C$. The stepsize γ_t also has a closed solution: $\min\{1, \frac{\langle \boldsymbol{d}_t, -\nabla f(\boldsymbol{\alpha}_t)\rangle}{\langle \boldsymbol{d}_t, \mathcal{K}\boldsymbol{d}_t\rangle}\}$. We leave the details to the appendix.

Weston–Watkins Multiclass SVM The Weston–Watkins (WW) [15] formulation is closely related to Crammer–Singer but uses separate slack variables per class instead of one per sample. The Weston–Watkins primal is:

$$\min_{W,\xi} \frac{1}{2}\sum_{k=1}^{K}\|w_k\|_2^2 + C\sum_{i=1}^{n}\sum_{\substack{k=1\\k\neq y_i}}^{K}\xi_{i,k},\ \text{s.t. } \forall k \neq y_i, w_{y_i}^\top x_i - w_k^\top x_i \geq 1 - \xi_{i,k},\ \xi_{i,k} \geq 0.$$

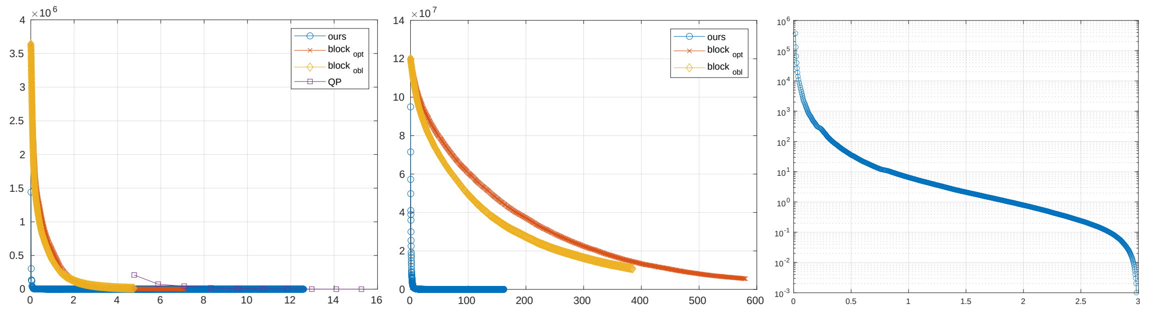

Figure 1: Left figure is on DNA dataset, where $n = 2000, K = 3, d = 180$. We compare various methods ($block_{obl}, block_{opt}$ denotes 'oblivious' and 'optimal' stepsize in [9], repectively) to solve **Crammer–Singer** formulation. Similar results for **Weston–Watkins** version. X-axis denotes time consumption (in seconds) while Y-axis denotes objective of $f(\boldsymbol{\alpha}_t) - f^*$ in Eq. (9). Middle figure is on USPS dataset where $n = 7291, K = 10, d = 256$. We ignore QP due to out of memory issue, we can see our method is way faster than [9]. Right figure is our proposed L2-SVM method on *krvskp* dataset ($n = 3196, K = 2, d = 36$), which converges very fast. All datasets can be found in [5].

The dual objective is exactly the same as Eq. (9), while the constraint is a little different: $(\boldsymbol{\alpha}\mathbf{1} = \mathbf{0}) \wedge (\forall i, \forall j \neq y_i, -C \leq \boldsymbol{\alpha}_{ij} \leq 0)$. (Please refer to Appendix A.) Therefore, we can still use Algorithm 2 to solve it, except the first line in the loop is a little different. It is to compare $\boldsymbol{q}(y_i)$ with each of the remaining element: for $j \neq y_i$, if $\boldsymbol{q}(y_i) < \boldsymbol{q}(j)$, then $\boldsymbol{s}_t(i,j) = -C$, otherwise $\boldsymbol{s}_t(i,j) = 0$. $\boldsymbol{s}_t(i, y_i)$ can be determined as $-\sum_j \boldsymbol{s}_t(i,j)$ given the equality constraint. One can see when $\boldsymbol{q}(y_i) == \max(\boldsymbol{q})$, WW update is exactly the same as CS version, both yielding $\mathbf{0}$.

Theorem 1 *For either WW or CS SVMs, our proposed method has sublinear linear convergence rate. Specifically, for CS we have* $f(\boldsymbol{\alpha}_t) - f^* \leq \frac{4nC^2\sigma_1(\mathcal{K})}{t}$ *while for WW:* $f(\boldsymbol{\alpha}_t) - f^* \leq \frac{2nK(K-1)C^2\sigma_1(\mathcal{K})}{t}$. *By comparison, randomized FW proposed in [9] for CS can guarantee* $\mathbb{E}(f(\boldsymbol{\alpha}_t)) - f^* \leq \frac{4nC^2\|\mathcal{K}\|_{2,1}}{t+2n-2}$ *while for WW:* $\mathbb{E}(f(\boldsymbol{\alpha}_t)) - f^* \leq \frac{2nK(K-1)C^2\|\mathcal{K}\|_{2,1}}{t+2n-2}$ *(See Appendix C).*

Theorem 2 *While our method can achieve sublinear convergence rate* $\mathcal{O}(1/t)$*, variants such as away-steps FW, pairwise FW and fully-corrective FW may accelerate convergence rate [8].*

5. Experiments

We first notice that Eq. (9) can also be solved via QP, where the objective can be reformulated as $\frac{1}{2}\text{vec}^\top(\boldsymbol{\alpha})(\boldsymbol{I} \otimes \mathcal{K})\text{vec}(\boldsymbol{\alpha}) + \text{vec}^\top(\boldsymbol{\alpha})(-\text{vec}(\mathbf{1}_{ind}))$. Since the size of $\boldsymbol{I} \otimes \mathcal{K}$ is as large as $Kn \times Kn$ and the complexity is cubic level, we found when $n = 2000, K = 10$, it runs for a long time while when $n = 5000, K = 20$, QP runs out of memory (125G). It is worth noting that for QP, storage can be a very practical issue. That said, in our experiments, for QP, we use the dataset where the size works. We also compare our matrix-wise update with stochastic block version (row wise) proposed in [9]. As illustrated in Fig. 1, our method is significantly faster than the rest. We also find that in the cases where QP yields lower objective than ours, the calssification accuracy barely changes, which indicates our method is extremely fast, and meanwhile can achieve good classification results.

References

[1] Francis Bach. *Learning theory from first principles*. MIT press, 2024.

[2] Amir Beck. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. SIAM, 2014.

[3] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[4] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[6] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[7] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

[8] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. *Advances in neural information processing systems*, 28, 2015.

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

[10] Ching-Pei Lee and Chih-Jen Lin. A study on l2-loss (squared hinge-loss) multiclass svm. *Neural computation*, 25(5):1302–1323, 2013.

[11] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[12] Fabian Pedregosa, Geoffrey Negiar, Armin Askari, and Martin Jaggi. Linearly convergent frank-wolfe with backtracking line-search. In *International conference on artificial intelligence and statistics*, pages 1–10. PMLR, 2020.

[13] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[15] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Technical Report CSD-TR-98-04, Department of Computer Science, Royal . . . , 1998.

[16] Qiang Wu and Ding-Xuan Zhou. Svm soft margin classifiers: linear programming versus quadratic programming. *Neural computation*, 17(5):1160–1187, 2005.

Appendix A. Dual of Crammer–Singer Multiclass SVM

We now turn to derive the dual for Crammer–Singer Multiclass SVM in Eq. (9). For each sample x_i, y_i: since $w_{y_i}^\top \phi(x_i) - w_k^\top \phi(x_i) \geq 1 - \xi_i$, we introduce Lagrangian Multiplier $\boldsymbol{\alpha}_{ik}$ $(k \neq y_i)$ while $\xi_i \geq 0$, we introduce λ_i.

$$L(W,\xi;\lambda,\boldsymbol{\alpha}) = \frac{1}{2}\sum_{k=1}^{K}\|w_k\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_i\sum_{k\neq y_i}\boldsymbol{\alpha}_{ik}(w_{y_i}^\top\phi(x_i) - w_k^\top\phi(x_i) - 1 + \xi_i) - \sum_i \lambda_i\xi_i \tag{10}$$

After taking the derivative with respect to W, ξ, we have $W = \Phi(X)\boldsymbol{\alpha}$ where i-th row of $\boldsymbol{\alpha}$ is $[-\boldsymbol{\alpha}_{i1}, -\boldsymbol{\alpha}_{i2}, \ldots, -\boldsymbol{\alpha}_{i(y_i-1)}, \sum_{k\neq y_i}\boldsymbol{\alpha}_{ik}, -\boldsymbol{\alpha}_{i(y_i+1)}, \ldots, -\boldsymbol{\alpha}_{iK}]$ and $\lambda_i + \sum_{k\neq y_i}\boldsymbol{\alpha}_{ik} = C$ where the second is equivalent to $0 \leq \sum_{k\neq y_i}\boldsymbol{\alpha}_{ik} \leq C$. Given $W = \Phi(X)\boldsymbol{\alpha}$, we then have

$$\|W\|_F^2 = \text{trace}(\boldsymbol{\alpha}^\top\Phi(X)^\top\Phi(X)\boldsymbol{\alpha}) = \text{trace}(\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha}). \tag{11}$$

With simple algebraic operations, one can obtain Eq. (9) as desired. **Weston–Watkins** formulation is more or less the same except more multipliers will be introduced:

$$L(W,\xi;\lambda,\boldsymbol{\alpha}) = \frac{1}{2}\sum_{k=1}^{K}\|w_k\|^2 + C\sum_i\sum_{k\neq y_i}\xi_{i,k} - \sum_i\sum_{k\neq y_i}\boldsymbol{\alpha}_{ik}(w_{y_i}^\top\phi(x_i) - w_k^\top\phi(x_i) - 1 + \xi_{i,k}) - \sum_i\sum_{k\neq y_i}\lambda_{i,k}\xi_{i,k}. \tag{12}$$

By taking the derivative w.r.t. $\xi_{i,k}$ and set it to 0, we obtain $C = \boldsymbol{\alpha}_{ik} + \lambda_{i,k}$, implying $0 \leq \boldsymbol{\alpha}_{ik} \leq C$.

Appendix B. Closed-solution Stepsize

By taking the derivative of $f(\boldsymbol{\alpha}_t + \gamma\boldsymbol{d}_t)$ with repect to γ, we can obtain $\gamma = \frac{\langle \boldsymbol{d}_t, -\nabla f(\boldsymbol{\alpha}_t)\rangle}{\langle \boldsymbol{d}_t, \mathcal{K}\boldsymbol{d}_t\rangle}$. One can prove that the numerator is always nonnegative [3, 12]. Therefore, given the constraint of $\gamma \in [0,1]$, we conclude $\gamma = \min\{1, \frac{\langle \boldsymbol{d}_t, -\nabla f(\boldsymbol{\alpha}_t)\rangle}{\langle \boldsymbol{d}_t, \mathcal{K}\boldsymbol{d}_t\rangle}\}$. Similar optimal stepsize can also be obtained for block-wise update in [9]: $\gamma = \min\{1, \frac{\langle \boldsymbol{e}_i\boldsymbol{d}_t, -\nabla f(\boldsymbol{\alpha}_t)\rangle}{\langle \boldsymbol{e}_i\boldsymbol{d}_t, \mathcal{K}\boldsymbol{e}_i\boldsymbol{d}_t\rangle}\}$, where $\boldsymbol{d}_t$ is a **row vector** since block-wise is updated row by row in this specific problem and $\boldsymbol{e}_i$ is a column zero vector except i-th element is 1. In the left part of Fig. 1, we use above stepsize for *block*$_{opt}$ while $2n/(t+2n)$ as *block*$_{obl}$ stepsize.

Appendix C. Theorem 1

This section largely follows [3], from which we introduce the following lemma:

Lemma 3 *Let p be a positive integer, and let $\{a_k\}_{k\geq 0}$ and $\{b_k\}_{k\geq 0}$ be nonnegative sequences satisfying for any $k \geq 0$*

$$a_{k+1} \leq a_k - \gamma_k b_k + \frac{A}{2}\gamma_k^2,$$

where $\gamma_k = \frac{2}{k+2p}$ and A is a positive constant. Suppose that $a_k \leq b_k$ for all k. Then:

(a) For any $k \geq 1$,

$$a_k \leq \frac{2\max\{A, (p-1)a_0\}}{k+2p-2}.$$

(b) For any $k \geq 3$,

$$\min_{n=\lfloor k/2 \rfloor+2,\ldots,k} b_n \leq \frac{8\max\{A,(p-1)a_0\}}{k-2}.$$

Theorem 4 *For $\min F(x) = f(x) + g(x)$ where g is convex and its domain is compact, while f is convex. Let $\{x_k\}_{k\geq 0}$ be the sequence generated by conditional gradient method for optimizing F with exact line search. Let Ω be an upper bound on the diameter of $\mathrm{dom}(g)$:*

$$\Omega \geq \max_{x,y\in \mathrm{dom}(g)} \|x-y\|,$$

and define $S(x) = \langle \nabla f(x), x-p\rangle + g(x) - g(p)$ where where $p \in \arg\min_{\boldsymbol{s}\in\mathcal{C}} \langle \nabla f(x), \boldsymbol{s}\rangle$. Then:

(a) $F(x_k) - F_{opt} \leq \dfrac{2L_f\Omega^2}{k}$ for any $k \geq 1$;

(b) $\min_{n=\lfloor k/2\rfloor+2,\ldots,k} S(x_n) \leq \dfrac{8L_f\Omega^2}{k-2}$ for any $k \geq 3$.

Proof By the inequality (13.6) [3] invoked with $x = x_k$, $t = t_k$ and g denoting the indicator function whether it is in the feasible domain, we have for any $k \geq 0$:

$$F(x_k + t_k(p_k - x_k)) - F_{\text{opt}} \leq F(x_k) - F_{\text{opt}} - t_k S(x_k) + \frac{t_k^2 L_f}{2}\|p_k - x_k\|^2.$$

For $t_k = \alpha_k = \frac{2}{k+2}$ (predefined),

$$F(x_k + \alpha_k(p_k - x_k)) - F_{\text{opt}} \leq F(x_k) - F_{\text{opt}} - \alpha_k S(x_k) + \frac{\alpha_k^2 L_f}{2}\|p_k - x_k\|^2.$$

For exact line search ($t_k = u_k$), by definition:

$$F(x_k+u_k(p_k-x_k))-F_{\text{opt}} \leq F(x_k+\alpha_k(p_k-x_k))-F_{\text{opt}} \leq F(x_k)-F_{\text{opt}}-\alpha_k S(x_k)+\frac{\alpha_k^2 L_f}{2}\|p_k-x_k\|^2.$$

Using $\|p_k - x_k\| \leq \Omega$, we get:

$$F(x_{k+1}) - F_{\text{opt}} \leq F(x_k) - F_{\text{opt}} - \alpha_k S(x_k) + \frac{\alpha_k^2 L_f \Omega^2}{2}.$$

Apply Lemma 3 with $a_k = F(x_k) - F_{\text{opt}}$, $b_k = S(x_k)$, $A = L_f\Omega^2$, and $p = 1$. Also, Lemma 13.12 in [3] ensures $a_k \leq b_k$. By parts (a) and (b) of Lemma 3, we obtain the desired result. ■

For each row of $\boldsymbol{\alpha}$ in CS, given the constraint in Eq. (9), we have $\|\alpha[i,:]-\hat{\alpha}[i,:]\|^2 \leq 2C^2$ where the equality holds when taking extreme points $\{(C,-C,0,\ldots,0),(C,0,-C,\ldots,0),\ldots,(0,0,0,\ldots,0)\}$, therefore we have $\Omega^2 = 2nC^2$. For WW, one can see for each row, the extreme points change to the origin plus $((K-1)C,-C,-C,\ldots,-C)$ and its permutations, where we can conclude $\Omega^2 = nK(K-1)C^2$. By making use the simple fact that $L_f = \sigma_1(\mathcal{K})$, we obtain the desired result in the first part of Theorem 1.

We now turn to stochastic block version, and consider the problem:

$$\min_{x_1\in E_1,\ldots,x_p\in E_p}\left\{F(x_1,\ldots,x_p):=f(x_1,\ldots,x_p)+\sum_{j=1}^{p}g_j(x_j)\right\}, \tag{13}$$

where $E_1,\ldots,E_p$ are Euclidean spaces and the product space is defined as $E := E_1\times E_2\times\cdots\times E_p$, equipped with the norm $\|(u_1,\ldots,u_p)\|_E := \left(\sum_{i=1}^p \|u_i\|^2\right)^{1/2}$. Define: $g(x_1,\ldots,x_p) := \sum_{i=1}^p g_i(x_i)$, with $\text{dom}(g)=\text{dom}(g_1)\times\cdots\times\text{dom}(g_p)$. The gradient of f with respect to block $i\in\{1,\ldots,p\}$ is denoted by $\nabla_i f$, and we have: $\nabla f(x)=(\nabla_1 f(x),\nabla_2 f(x),\ldots,\nabla_p f(x))$. For each i, define the linear operator $U_i : E_i\to E$ as: $U_i(d):=(0,\ldots,0,\underbrace{d}_{i\text{-th block}},0,\ldots,0),\quad d\in E_i$.

We write $x=(x_1,\ldots,x_p)\equiv(x_i)_{i=1}^p$, so problem (13) becomes:

$$\min_{x\in E}\{F(x):=f(x)+g(x)\}.$$

We make the following assumptions:

(A) Each $g_i : E_i\to(-\infty,\infty]$ is proper, closed, convex, and has compact domain.

(B) $f : E\to(-\infty,\infty]$ is convex and differentiable over $\text{dom}(f)$, and $\text{dom}(g)\subseteq\text{dom}(f)$.

(C) For each i, there exists $L_i>0$ such that for all $x\in\text{dom}(f)$ and $d\in E_i$ with $x+U_i(d)\in\text{dom}(f)$:

$$\|\nabla_i f(x)-\nabla_i f(x+U_i(d))\|\le L_i\|d\|.$$

(D) The optimal set X^* is nonempty and the optimal value is F_{opt}.

and make the following **notations:** $\xi_{k-1}=\{i_0,\ldots,i_{k-1}\}$ denotes the history of block choices; weighted norm: $\|x\|_L := \left(\sum_{i=1}^p L_i\|x_i\|^2\right)^{1/2}$.

Algorithm 3: Block-wise FW (update row by row) to solve Eq. (9)

```
Data: Given $C$, $(x_i, y_i)_{i=1}^n$, formulate $\mathcal{K}$ and $\mathbf{1}_{ind}$, and initialize $t=0, \boldsymbol{\alpha}=\mathbf{0}$
Result: Obtain optimal $\boldsymbol{\alpha}$ to Eq. (9)
while $k \le T$ do
    Sample $i_k \in \{1,2,\ldots,n\}$ uniformly at random;
    choose step size $t_k = 2n/(k+2n)$ or via exact linear search;
    Update: $x^{k+1} = x^k + t_k U_{i_k}(p_{i_k} - x^k_{i_k})$, where $p_{i_k} \in \arg\min_{v\in E_{i_k}} g_{i_k}(v) + \langle v, \nabla_{i_k} f(x^k)\rangle$;
    $t = t+1$;
end
```

Theorem 5 *Suppose that the above assumptions hold, and let $\{x^k\}_{k\ge 0}$ be the sequence generated by the block-wise FW method (applicable to Eq. (9)) with stepsize $t_k=\frac{2p}{k+2p}$. Let Ω satisfy*

$$\Omega\ge\max_{x,y\in\text{dom}(g)}\|x-y\|_L. \tag{14}$$

Then:

(a) For any $k \geq 1$,

$$\mathbb{E}_{\xi_{k-1}}[F(x^k)] - F_{opt} \leq \frac{2\max\{(p-1)(F(x^0)-F^*),\ p\Omega^2\}}{k+2p-2}. \tag{15}$$

(b) For any $k \geq 3$,

$$\min_{n=\lfloor k/2\rfloor+2,\ldots,k} \mathbb{E}_{\xi_{n-1}}[S(x^n)] \leq \frac{8\max\{(p-1)(F(x^0)-F^*),\ p\Omega^2\}}{k-2}. \tag{16}$$

Proof Let $p^k = p(x^k)$, and note that $p_i^k = p_i(x^k)$. From the block descent lemma and convexity of g_{i_k} (indicator function for the constraint of each row, which is convex), we have:

$$\begin{aligned}
&F(x^{k+1}) = f(x^k + t_k U_{i_k}(p_{i_k}^k - x_{i_k}^k)) + g(x^k + t_k U_{i_k}(p_{i_k}^k - x_{i_k}^k))\\
\leq& f(x^k) - t_k\langle \nabla_{i_k} f(x^k), x_{i_k}^k - p_{i_k}^k\rangle + \frac{t_k^2 L_{i_k}}{2}\|p_{i_k}^k - x_{i_k}^k\|^2 + \sum_{j\neq i_k} g_j(x_j^k) + g_{i_k}((1-t_k)x_{i_k}^k + t_k p_{i_k}^k)\\
=& f(x^k) - t_k\langle \nabla_{i_k} f(x^k), x_{i_k}^k - p_{i_k}^k\rangle + \frac{t_k^2 L_{i_k}}{2}\|p_{i_k}^k - x_{i_k}^k\|^2 + g(x^k) - g_{i_k}(x_{i_k}^k) + g_{i_k}((1-t_k)x_{i_k}^k + t_k p_{i_k}^k)\\
\leq& F(x^k) - t_k S_{i_k}(x^k) + \frac{t_k^2 L_{i_k}}{2}\|p_{i_k}^k - x_{i_k}^k\|^2.
\end{aligned}$$

Taking expectation w.r.t. i_k, which is sampled uniformly:

$$\mathbb{E}_{i_k}[F(x^{k+1})] \leq F(x^k) - \frac{t_k}{p}\sum_{i=1}^{p} S_i(x^k) + \frac{t_k^2}{2p}\sum_{i=1}^{p} L_i\|p_i^k - x_i^k\|^2 = F(x^k) - \frac{t_k}{p}S(x^k) + \frac{t_k^2}{2p}\|p^k - x^k\|_L^2.$$

Now taking expectation over ξ_{k-1}, defining $\alpha_k = \frac{t_k}{p} = \frac{2}{k+2p}$ and using the definition $\|x - y\|_L \leq \Omega$, we get:

$$\mathbb{E}_{\xi_k}[F(x^{k+1})] \leq \mathbb{E}_{\xi_{k-1}}[F(x^k)] - \alpha_k \mathbb{E}_{\xi_{k-1}}[S(x^k)] + \frac{p\alpha_k^2}{2}\Omega^2.$$

By subtracting F^* from both sides, we get:

$$\mathbb{E}_{\xi_k}[F(x^{k+1})] - F^* \leq \mathbb{E}_{\xi_{k-1}}[F(x^k)] - F^* - \alpha_k \mathbb{E}_{\xi_{k-1}}[S(x^k)] + \frac{p\alpha_k^2}{2}\Omega^2.$$

By invoking Lemma 3 with $a_k = \mathbb{E}_{\xi_{k-1}}[F(x^k)] - F^*$, $b_k = \mathbb{E}_{\xi_{k-1}}[S(x^k)]$ and noticing the simple fact that $a_k \leq b_k$ (definition of S), we obtain the desired result. ■

By replacing x with $\boldsymbol{\alpha}$ for Eq. (9), we can get the bound for stochastic block-wise update. We now turn to bound Ω. By definition, $\|\nabla_i f(\boldsymbol{\alpha}) - \nabla_i f(\boldsymbol{\alpha} + U_i(d))\| = \|\mathcal{K}[:,i]U_i(d)\| = \|\mathcal{K}[:,i]\|\|d\|$, which indicates $L_i = \|\mathcal{K}[:,i]\|$. Thus for CS: $\Omega^2 = \sum_{i=1}^{n} L_i * 2C^2 = 2\|\mathcal{K}\|_{2,1}C^2$; while for WW: $\Omega^2 = \sum_{i=1}^{n} L_i * K * (K-1)C^2 = K(K-1)\|\mathcal{K}\|_{2,1}C^2$, we obtain the desired result in the second half of Theorem 1.