Exact Instance Compression for Convex Empirical Risk Minimization via Color Refinement

Bryan Zhu
Massachusetts Institute of Technology
Cambridge, MA 02139
bryanzhu@mit.edu

Ziang Chen
Massachusetts Institute of Technology
Cambridge, MA 02139
ziang@mit.edu

Abstract

Empirical risk minimization (ERM) can be computationally expensive, with standard solvers scaling poorly even in the convex setting. We propose a novel lossless compression framework for convex ERM based on color refinement, extending prior work from linear programs and convex quadratic programs to general differentiable convex optimization problems. Constructing such a coloring is objective-specific, and we do not claim an automatic procedure for arbitrary convex ERM. We develop concrete, constructive algorithms realizing this framework for a range of models, including linear and polynomial regression, binary and multiclass logistic regression, regression with elastic-net regularization, and kernel methods such as kernel ridge regression and kernel logistic regression. Numerical experiments on representative datasets demonstrate the effectiveness of the proposed approach.

1 Introduction

Empirical risk minimization (ERM) is a foundational paradigm in modern machine learning and statistical inference, underlying a wide range of models, including linear and generalized linear models, kernel methods, and regularized regression.

Although many ERM formulations are convex, solving them at scale remains computationally challenging: the complexity of first- and second-order optimization methods typically grows at least linearly with the number of samples and features.

These computational bottlenecks have motivated extensive research on preprocessing techniques that seek to reduce instance size while preserving optimal solutions, including randomized sketching [Mahoney, 2011, Pilanci and Wainwright, 2016], subsampling [Drineas et al., 2012], coreset construction [Feldman et al., 2020], and symmetry-based reductions [Ravanbakhsh et al., 2017].

In this paper, we study lossless compression of convex ERM problems. Unlike approximate methods, lossless compression yields a reduced optimization problem whose solution can be lifted exactly to a solution of the original problem. These approaches are complementary: exact reduction eliminates redundant structure without loss of optimality, after which approximate methods may be applied to the reduced problem. Thus, exact reduction is not a direct alternative to approximation, but operates at a different level.

Our approach builds on recent advances in dimension reduction for optimization based on color refinement, a classical algorithm from graph isomorphism theory [Weisfeiler and Leman, 1968]. Color refinement iteratively assigns colors to vertices of a graph in a way that captures structural equivalence. Since matrices can be naturally represented as edge-weighted bipartite graphs, color refinement can be used to identify functionally equivalent rows and columns in matrices in optimization problems.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Recent work has shown that this idea leads to exact dimension reduction for linear programs (LPs), where variables and constraints can be grouped into equivalence classes [Grohe et al., 2014]. This framework was subsequently extended to convex quadratic programs (QPs) [Mladenov et al., 2017].

However, existing methods rely on linear or quadratic structure and do not directly apply to general convex ERM objectives, such as those arising from logistic loss, multinomial loss, elastic-net regularization, or kernelized models. Moreover, prior symmetry-based approaches typically require explicit permutation invariance of the problem, which is often absent in practical machine learning formulations even when many samples or features behave identically.

In this work, we develop a general dimension reduction theorem for differentiable convex programming and instantiate it in practical algorithms for convex ERM families. Our method identifies variables and constraints that behave identically under objective and constraint gradients on color-constant assignments, even when the optimization problem lacks explicit permutation symmetry.

In practice, however, exact reductions are most effective for sparse, categorical, quantized, or repeated-pattern data, where exact structural redundancy naturally arises, and offer little benefit on generic dense continuous data. Empirically, we observe that our method can reduce problem size and training cost in structured settings.

The remainder of the paper is organized as follows. Section 2 presents our generalization of color-refinement–based reductions to convex programs. Section 3 discusses applications of our theorem to machine learning models, including linear and polynomial regression, binary and multiclass logistic regression, regression with elastic-net regularization, and kernel-based models such as kernel ridge regression and kernel logistic regression. Section 4 reports experimental results for binary logistic regression, multiclass logistic regression, kernel binary logistic regression, ridge regression, and kernel ridge regression on datasets from OpenML [Feurer et al., 2021]. Section 5 concludes the paper.

2 Dimension Reduction

2.1 Preliminaries

We consider a general convex program with n variables and m constraints:

$$\min \quad F(x), \;\; \text{subject to} \quad G_i(x) \le b_i, \quad i = 1, \dots, m, \quad l \le x \le u.$$

We assume that F and each G_i are convex and differentiable on the *box domain*, $\{x \mid l \le x \le u\}$. We call x *feasible* if it lies within the box domain and satisfies all constraints $G_i(x) \le b_i$.

The key idea of reduction is that when the convex program exhibits sufficient regularity across variables and constraints, many of them can be aggregated without loss of optimality. This aggregation is formalized through the notion of equitable partitions.

We say that a *partition* of a set S is a set of nonempty, pairwise disjoint subsets of S whose union equals S. Let $\mathcal{P}$ be a partition of the row indices of a matrix A, and let $\mathcal{Q}$ be a partition of its column indices. We say that the pair $(\mathcal{P}, \mathcal{Q})$ is *equitable* if, for every $S \in \mathcal{P}$ and $T \in \mathcal{Q}$, $\sum_{j \in T} A_{ij}$ is constant over all $i \in S$, and $\sum_{i \in S} A_{ij}$ is constant over all $j \in T$. Equivalently, each submatrix induced by $\mathcal{P} \times \mathcal{Q}$ has both constant row sums and constant column sums.

Equitable partitions can be computed via *color refinement*, a graph isomorphism algorithm [Berkholz et al., 2017]. It gradually refines a coloring of a graph, which can also be seen as a partition of its vertices. A coloring $\mathcal{C}_1$ *refines* another coloring $\mathcal{C}_2$ if, for all $C_1 \in \mathcal{C}_1$, there exists some $C_2 \in \mathcal{C}_2$ such that $C_1 \subseteq C_2$. Here, C_1 is *finer*, and C_2 is *coarser*. Additionally, we call $\mathcal{C}$ the *unit coloring* if $|\mathcal{C}| = 1$.

Color refinement can be extended to matrices to find the equitable partition $(\mathcal{P}, \mathcal{Q})$ that minimizes $|\mathcal{P}|$ and $|\mathcal{Q}|$, which can be shown to be unique [Grohe et al., 2014]. This is called the *coarsest equitable partition*. We will present an explicit version in Section 2.4.

To express aggregation algebraically, for any partition $\mathcal{P}$ of a finite set V, define the *partition matrix* $\Pi_{\mathcal{P}} \in \{0,1\}^{V \times \mathcal{P}}$ by

$$(\Pi_{\mathcal{P}})_{vS} = \begin{cases} 1, & v \in S, \\ 0, & \text{otherwise.} \end{cases}$$

We also define the scaled transpose $\Pi_{\mathcal{P}}^{\text{Scaled}}$ by normalizing each row of $\Pi_{\mathcal{P}}^{\top}$ so that it is stochastic.

In Appendix A, we will review prior work in more detail and describe how equitable partitions yield exact reductions for linear and quadratic programs.

We now generalize this principle to differentiable convex empirical risk minimization (ERM).

2.2 Reduction Theorem

Definition 2.1 (Reduction Coloring)**.** A coloring $\mathcal{P}$ of the constraints and a coloring $\mathcal{Q}$ of the variables of a convex program is a *reduction coloring* if the following conditions hold for all $\hat{x}$ in the box domain satisfying $\hat{x}_{j_1} = \hat{x}_{j_2}$ whenever j_1 and j_2 belong to the same color in $\mathcal{Q}$:

- If $T \in \mathcal{Q}$, then for all $j_1, j_2 \in T$,
$$\left.\frac{\partial F}{\partial x_{j_1}}\right|_{x=\hat{x}} = \left.\frac{\partial F}{\partial x_{j_2}}\right|_{x=\hat{x}}.$$
- If $S \in \mathcal{P}$ and $T \in \mathcal{Q}$, then for all $j_1, j_2 \in T$,
$$\left.\frac{\partial}{\partial x_{j_1}}\left(\sum_{i\in S} G_i\right)\right|_{x=\hat{x}} = \left.\frac{\partial}{\partial x_{j_2}}\left(\sum_{i\in S} G_i\right)\right|_{x=\hat{x}}.$$
- For each $S \in \mathcal{P}$, $G_i(\hat{x})$ is equal for every $i \in S$.
- For each $S \in \mathcal{P}$, b_i is equal for every $i \in S$.
- For each $T \in \mathcal{Q}$, (l_j, u_j) is equal for every $j \in T$.

In general convex programs, verifying these conditions directly may be nontrivial. In Section 3, we provide constructive sufficient conditions for common ERM families, where reduction colorings are computed efficiently via color refinement applied to the data matrix, labels, and weights.

Definition 2.2. Given a reduction coloring $(\mathcal{P}, \mathcal{Q})$ of a convex program, we define a *reduced program*:

- The reduced objective F' is obtained by substituting a single reduced variable x'_T for all original variables x_j belonging to the same $T \in \mathcal{Q}$. In other words, for each reduced vector $x' \in \mathbb{R}^{|\mathcal{Q}|}$, we set a point $\hat{x} \in \mathbb{R}^n$ such that $\hat{x}_j = x'_T$ whenever $j \in T$. Then we set $F'(x') = F(\hat{x})$.
- Reduced constraints G'_S are given by transforming G_i similarly as the objective for any $i \in S$ (well defined since $G'_S(x') = G_i(\hat{x})$ is equal for all $i \in S$ for all color-constant $\hat{x}$).
- For x, l, and u, we left-multiply by $\Pi_{\mathcal{Q}}^{\text{Scaled}}$ to yield the reduced vectors.
- For b, we use $b' = \Pi_{\mathcal{P}}^{\text{Scaled}} b$.

We can now state the main reduction theorem.

Theorem 2.3. *Consider any convex program with F and each G_i differentiable on the box domain. Let $(\mathcal{P}, \mathcal{Q})$ be a reduction coloring. If x is an optimum for the original program, then $x' = \Pi_{\mathcal{Q}}^{\textit{Scaled}} x$ is an optimum for the reduced program. If x' is an optimum for the reduced program, then $x = \Pi_{\mathcal{Q}} x'$ is an optimum for the original program.*

Informally, we reduce by averaging every variable in a color down to one shared value and solving the smaller problem, and we lift by copying each reduced value back out to every original variable that shares its color.

The proof proceeds in three steps. First, averaging within color classes cannot increase the objective under the gradient consistency assumptions. Second, feasibility is preserved under the same averaging operation by applying the argument to the constraint functions. Finally, the original problem and the reduced problem are shown to have identical optimal values, and optimal solutions lift exactly between the two formulations. A full proof is given in Appendix C.

2.3 Beyond Permutation Symmetry

Reduction colorings capture equivalence under aggregated interactions, rather than requiring explicit permutation invariance of the optimization problem. As a result, our framework can yield strictly stronger compression than approaches based solely on symmetry groups.

Formally, let Γ denote the group of automorphisms $(\pi, \sigma) \in S_n \times S_m$ such that for all x in the box domain:

- $F(\pi(x)) = F(x)$;
- $\pi(l) = l$ and $\pi(u) = u$;
- $G_{\sigma(i)}(\pi(x)) = G_i(x)$ and $b_{\sigma(i)} = b_i$ for all $i = 1, \dots, m$.

The group Γ acts on variables by π and on constraints by σ. Let $\mathcal{Q}_\Gamma$ and $\mathcal{P}_\Gamma$ denote the corresponding orbit partitions of variables and constraints.

Theorem 2.4. *If $(\mathcal{P}, \mathcal{Q})$ is the coarsest reduction coloring, then it is at least as coarse as $(\mathcal{P}_\Gamma, \mathcal{Q}_\Gamma)$.*

The proof, given in Appendix D, shows that any permutation symmetry induces a valid reduction coloring, implying that reduction colorings strictly generalize symmetry-based reductions. We also provide an example where nontrivial compression is possible even in the absence of any nontrivial permutation symmetry.

2.4 Color Refinement for Matrices

While matrix equitability does not explicitly appear in Theorem 2.3, equitability is often required on the data or kernel matrix for convex ERM instantiations of our compression method. Below, we present the algorithm we use for refining a coloring $(\mathcal{P}, \mathcal{Q})$ of a matrix A until it converges to the coarsest equitable partition. This is essentially a weighted case of color refinement for graphs [Berkholz et al., 2017].

Algorithm 1: Color refinement for coarsest equitable partition of a matrix

Initialize: $\mathcal{P}$ and $\mathcal{Q}$ as needed, depending on the specific case
Add all colors in $\mathcal{P} \cup \mathcal{Q}$ to a stack S_{refine}
while $S_{\text{refine}} \neq \emptyset$ **and** ($|\mathcal{P}| < m$ **or** $|\mathcal{Q}| < n$) **do**
    $R \leftarrow \textsc{Pop}(S_{\text{refine}})$
    if $R \in \mathcal{P}$ **then**
        for $T \in \mathcal{Q}$ **do**
            Refine $\mathcal{Q}$: partition all $j \in T$ by $\left\{\sum_{i \in R} A_{ij}\right\}$.
        end for
        Push all newly created colors into S_{refine}
    else if $R \in \mathcal{Q}$ **then**
        for $S \in \mathcal{P}$ **do**
            Refine $\mathcal{P}$: partition all $i \in S$ by $\left\{\sum_{j \in R} A_{ij}\right\}$.
        end for
        Push all newly created colors into S_{refine}
    end if
end while
Return $(\mathcal{P}, \mathcal{Q})$

When we split a color R into sub-colors $\mathcal{S} = \{\gamma, \gamma', \gamma'', \dots\}$ (we assume without loss of generality that $|\gamma| \geq |\gamma'| \geq |\gamma''| \geq \dots$), γ keeps the R label, while each of $\mathcal{S} \setminus \{\gamma\}$ is assigned a distinct new label. Hence, all sub-colors in $\mathcal{S} \setminus \{\gamma\}$ are pushed into S_{refine}.

Proposition 2.5. *The complexity of Algorithm 1 is $O(mn(\log(m) + \log(n)))$.*

In many ERM settings, A is sparse, meaning most entries are exactly 0 and A is stored by listing only its nonzero entries. In this case, refinement can be implemented by aggregating only nonzero entries with an index in the refining color, so the dense dependence on mn can be replaced by a dependence on $\text{nnz}(A)$, yielding a sparse implementation with runtime $O(\text{nnz}(A)(\log m + \log n))$.

We will prove Proposition 2.5 in Appendix E.

3 Applications in Machine Learning

We now instantiate the general reduction framework from Section 2 for several widely used convex ERM models. In each case, we derive explicit, verifiable conditions under which samples, features, or coefficients can be aggregated without loss of optimality.

3.1 Least Squares Linear Regression and Polynomial Regression

Consider the following least squares linear regression problem with n samples and D features:

$$\min\ F(w,b) = \|\hat{y} - y\|_2^2,$$

where $\hat{y} = Xw + b1_n$ is the prediction ($1_n \in \mathbb{R}^n$ is a vector of ones, so adding $b1_n$ adds the bias term $b \in \mathbb{R}$ to each row/sample of Xw), $X \in \mathbb{R}^{n \times D}$ contains the input samples, and $y \in \mathbb{R}^n$ contains the ground truth labels.

Theorem 3.1. *A coloring $\mathcal{Q}$ of the coefficients w is a reduction coloring for linear regression if the following conditions are met:*

- *Define $\mathcal{P}$ as the coloring given by partitioning all $1 \le i \le n$ by $\left\{ \left(T, \sum_{j \in T} X_{ij} \right) \;\middle|\; T \in \mathcal{Q} \right\}$. We require that $(\mathcal{P}, \mathcal{Q})$ is equitable on X.*
- *If j_1 and j_2 share a color in $\mathcal{Q}$, then $\sum_{i=1}^n X_{ij_1} y_i = \sum_{i=1}^n X_{ij_2} y_i$. In other words, $(X^\top y)_{j_1} = (X^\top y)_{j_2}$.*

The proof is given in Appendix F.1. We also verify that the closed-form solution of the reduced problem lifts exactly to a solution of the original regression.

In practice, these conditions can be enforced by initializing $\mathcal{P}$ as the unit coloring and $\mathcal{Q}$ by splitting features j according to $\sum_{i=1}^n X_{ij} y_i$, then computing the coarsest equitable partition of X refining $(\mathcal{P}, \mathcal{Q})$. The reduced regression problem is obtained by projecting $X' = X\Pi_{\mathcal{Q}}$.

In fact, since each sample i sharing a color in $\mathcal{P}$ has identical features after this operation, we can further merge and weigh these samples by projecting $X' = \Pi_{\mathcal{P}}^{\text{Scaled}} X \Pi_{\mathcal{Q}}$ and $y' = \Pi_{\mathcal{P}}^{\text{Scaled}} y$, with the weights matrix $W' = \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}}$ being given by $W'_{SS} = |S|$ for all $S \in \mathcal{P}$. This is because having $|S|$ identical samples x_S with possibly different y is equivalent to replacing all y in all such samples with their mean y'_S because the loss function that we are aiming to minimize differs by a constant.

Remark 3.2. This method can also be applied to polynomial regression and, more generally, to any fixed feature map $\phi(x)$ because these are simply linear regression with an expanded X. Moreover, our method also applies to weighted linear regression because weighted linear regression is equivalent to transforming the feature matrix $X \mapsto W^{1/2}X$, where W is the diagonal weights matrix, and transforming the ground truth $y \mapsto W^{1/2}y$.

Remark 3.3. Both conditions above require exact equality, not approximate closeness. For continuous or noisy targets y, equality will generically fail to hold, so little or no compression should be expected; this is consistent with the discussion in Appendix B. The analogous conditions for other models are exact in the same sense.

3.2 Binary Logistic Regression

We next consider the weighted binary logistic regression problem

$$\min F(w,b) = -\sum_{i=1}^n v_i (y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)),$$

where $v \in \mathbb{R}^n$ is the weights vector, $\hat{y} = \sigma(Xw + b1_n)$ is the predicted probability vector, σ represents the sigmoid function, and $y \in \{0,1\}^n$ are the ground truth labels.

Theorem 3.4. *A coloring $\mathcal{Q}$ of the coefficients w is a reduction coloring for binary logistic regression if the following conditions are met:*

- *Define $\mathcal{P}$ as we did in Theorem 3.1. We require that $(\mathcal{P}, \mathcal{Q})$ is equitable on X.*

- *If j_1 and j_2 share a color in $\mathcal{Q}$, then $\sum_{i=1}^n v_i X_{ij_1} y_i = \sum_{i=1}^n v_i X_{ij_2} y_i$.*
- *v_i is constant across all $i \in S$ for any $S \in \mathcal{P}$.*

The proof of Theorem 3.4 is given in Appendix F.2.

Similar to linear regression, we can satisfy all these conditions by initializing $\mathcal{P}$ to split $1 \le i \le n$ by v_i and $\mathcal{Q}$ to split $1 \le j \le D$ by $\sum_{i=1}^n v_i X_{ij} y_i$, then finding the coarsest equitable partition of X that refines $(\mathcal{P}, \mathcal{Q})$.

The reduction process for logistic regression is identical to that of linear regression. In F', the summation is over $S \in \mathcal{P}$ instead of $1 \le i \le n$. For the weights vector, we use $v' = \Pi_{\mathcal{P}}^\top v$. If we would like to have a binary ground truth, we can alternatively merge samples in each $S \in \mathcal{P}$ into up to two reduced weighted samples, one with label 1 and one with label 0.

3.3 Multiclass Logistic Regression

We now extend the reduction to multiclass logistic regression with K classes:

$$\min \; F(W, b) = -\sum_{i=1}^n v_i \log(\hat{y}_{iy_i}),$$

where $W \in \mathbb{R}^{D \times K}$, $b \in \mathbb{R}^K$, $\hat{y} = \mathrm{softmax}(XW + 1_n b^\top)$ (where softmax is applied along the class dimension) is the predicted probability matrix, and $y \in \{1, 2, \ldots, K\}^n$ are the ground truth labels. As with before, let $\mathcal{P}$ be a partition of the samples, and let $\mathcal{Q}$ be a partition of the features. We do not reduce the number of classes.

Theorem 3.5. *A reduction coloring for multiclass logistic regression is defined the same way as for binary logistic regression in Theorem 3.4 (including equitability on X and the requirement that v_i is constant across any $S \in \mathcal{P}$). The only difference is that we generalize our requirement of equal $\sum_{i=1}^n v_i X_{ij} y_i$ across $j \in T$ for any $T \in \mathcal{Q}$ by requiring equal $\sum_{i=1}^n v_i X_{ij} \cdot 1\{y_i = c\}$ for all $1 \le c \le K$.*

We prove Theorem 3.5 in Appendix F.3.

Within each class, the reduction process for multiclass logistic regression is exactly identical to that of binary logistic regression. This means that $W' = \Pi_{\mathcal{Q}}^{\text{Scaled}} W$, $b' = b$, $v' = \Pi_{\mathcal{P}}^\top v$, $\hat{y}' = \mathrm{softmax}(X'W' + 1_{|\mathcal{P}|} b^\top)$, $X' = \Pi_{\mathcal{P}}^{\text{Scaled}} X \Pi_{\mathcal{Q}}$, $y' \in \mathbb{R}^{|\mathcal{P}| \times K}$ is a probability matrix with $y'_{Sc} = \frac{\sum_{i \in S} 1\{y_i = c\}}{|S|}$, and

$$F'(W', b') = -\sum_{S \in \mathcal{P}} v'_S \sum_{c=1}^K y'_{Sc} \log(\hat{y}'_{Sc}).$$

Alternatively, to ensure a binary ground truth, we can merge samples in each $S \in \mathcal{P}$ into up to K reduced weighted samples, one per class.

3.4 Elastic-Net Regularization

Theorem 3.6. *Consider adding an elastic-net regularization term*

$$\lambda_2 \|w\|_2^2 + \lambda_1 \|w\|_1$$

to the objective of any convex ERM problem covered by a reduction coloring. Then the same reduction coloring remains valid for the regularized objective.

Proof. Theorem 2.3 does not apply directly because the ℓ_1 term is not differentiable everywhere. However, replacing w by its color-averaged version $\hat{w}$ cannot increase either $\|w\|_2^2$ or $\|w\|_1$, by Jensen's inequality applied within each color. Therefore, the regularization term does not increase under the same averaging operation used in the proof of Theorem 2.3, and the reduction remains valid. □

For example, we verify that the closed-form solution for ridge regression is preserved in Appendix F.4.

Explicitly, the reduced regularizer is

$$\lambda_2 \left(\sum_{T \in \mathcal{Q}} |T| (w'_T)^2 \right) + \lambda_1 \sum_{T \in \mathcal{Q}} |T| |w'_T|.$$

3.5 Note on StandardScaler

For models such as linear and logistic regression, it is often beneficial to apply StandardScaler from scikit-learn [Pedregosa et al., 2011] to improve numerical stability and conditioning. Given a data matrix $X \in \mathbb{R}^{n \times D}$, StandardScaler computes feature-wise empirical means $\mu \in \mathbb{R}^D$ and standard deviations $\sigma \in \mathbb{R}^D$, and transforms each sample $x \in \mathbb{R}^D$ as $x_{\text{sc}} = D_\sigma^{-1}(x - \mu)$, where $D_\sigma = \text{diag}(\sigma_1, \ldots, \sigma_D)$.

However, we empirically observe that applying StandardScaler prior to computing the coarsest reduction coloring typically produces a strictly finer coarsest equitable partition. Hence, we proceed as follows:

1. Compute the reduction coloring on the original data matrix X, producing a partition $(\mathcal{P}, \mathcal{Q})$. Using this coloring, form the reduced matrix X'.
2. Apply StandardScaler to X'. Let $\mu' \in \mathbb{R}^{|\mathcal{Q}|}$ and $\sigma' \in \mathbb{R}^{|\mathcal{Q}|}$ denote the feature means and standard deviations of X'. Train the desired model on $x'_{\text{sc}} = D_{\sigma'}^{-1}(x' - \mu')$. For example, in logistic regression, the learned parameters satisfy $z = (w'_{\text{sc}})^\top x'_{\text{sc}} + b'_{\text{sc}}$.
3. Convert the fitted parameters back to the (unscaled) reduced feature space. Substituting $x'_{\text{sc}} = D_{\sigma'}^{-1}(x' - \mu')$, we obtain $z = (w'_{\text{sc}})^\top D_{\sigma'}^{-1}(x' - \mu') + b'_{\text{sc}} = (D_{\sigma'}^{-1} w'_{\text{sc}})^\top x' + \left(b'_{\text{sc}} - (D_{\sigma'}^{-1} w'_{\text{sc}})^\top \mu'\right)$. Hence, the parameters in the unstandardized reduced space are $w' = D_{\sigma'}^{-1} w'_{\text{sc}}$ and $b' = b'_{\text{sc}} - (w')^\top \mu'$.
4. Lift w' to the original feature space using the column partition $\mathcal{Q}$. The intercept remains $b = b'$.

At inference time, predictions may be computed directly on the original (unstandardized) feature vectors x using $z = w^\top x + b$. No additional centering or normalization is required, since the effect of standardization has already been absorbed into the transformed parameters w and b. This procedure ensures that training on the standardized reduced data is exactly equivalent (up to solver tolerance) to training on a standardized version of the original data under the reduction induced by $(\mathcal{P}, \mathcal{Q})$. (We do not mean that σ'_T equals either σ_{j_1} or σ_{j_2} individually for $j_1, j_2 \in T$; in general it does not. Instead, the coefficients this procedure produces at the optimum, lifted back to the original feature space, are exactly the optimal coefficients one would get by standardizing the original X and training, then converting the resulting model to operate on unstandardized samples at inference.)

Remark 3.7. The equivalence above applies directly when no regularization term is present. With the elastic net regularization from Theorem 3.6, the reduced penalty is $\lambda_2 \left(\sum_{T \in \mathcal{Q}} |T| (w'_T)^2 \right) + \lambda_1 \sum_{T \in \mathcal{Q}} |T| |w'_T|$. Substituting $w'_T = \frac{(w'_{\text{sc}})_T}{\sigma'_T}$ shows that, to preserve the regularizer defined in the original unstandardized coordinates, the standardized reduced problem must use $\lambda_2 \left(\sum_{T \in \mathcal{Q}} \frac{|T|}{(\sigma'_T)^2} (w'_{\text{sc}})_T^2 \right) + \lambda_1 \sum_{T \in \mathcal{Q}} \frac{|T|}{\sigma'_T} |(w'_{\text{sc}})_T|$ rather than a uniform penalty.

3.6 Kernel-Based Models

Finally, we consider kernel methods. After applying the standard representer theorem-based reduction, training reduces to a convex program in coefficients $\alpha \in \mathbb{R}^n$ whose structure is determined by the symmetric positive semidefinite kernel matrix $K_{ij} = k(x_i, x_j)$ with $K \in \mathbb{R}^{n \times n}$. Theorem 2.3 applies directly to this finite problem. Here, we color α by $\mathcal{Q}$.

3.6.1 Kernel Ridge Regression

Training of weighted kernel ridge regression reduces to

$$\min \; F(\alpha, b) = (K\alpha + b1_n - y)^\top W (K\alpha + b1_n - y) + \lambda \alpha^\top K \alpha,$$

where $y \in \mathbb{R}^n$ is the ground truth and $W = \text{diag}(v)$ for sample weights $v \in \mathbb{R}^n$.

Theorem 3.8. *A coloring $\mathcal{Q}$ of α is a reduction coloring for kernel ridge regression under the following conditions:*

- *$(\mathcal{Q}, \mathcal{Q})$ is equitable on K.*
- *For each $T \in \mathcal{Q}$, $\sum_{i=1}^{n} v_i K_{ij} y_i$ is constant across $j \in T$.*
- *v_i is constant across all $i \in T$ for any $T \in \mathcal{Q}$.*

Note that this is nearly identical to the conditions for primal linear regression under Theorem 3.1 and, by extension, primal ridge regression. We prove Theorem 3.8 in Appendix F.5. The reduced problem uses two distinct projections of K: the regularizer depends only on α and uses the symmetric matrix $K'_{\text{quad}} = \Pi_{\mathcal{Q}}^{\top} K \Pi_{\mathcal{Q}}$, while the prediction term depends on $\hat{y} = K\alpha + b1_n$, which is constant within each color by equitability of $(\mathcal{Q}, \mathcal{Q})$ on K and is therefore represented by $K'_{\text{pred}} = \Pi_{\mathcal{Q}}^{\text{Scaled}} K \Pi_{\mathcal{Q}}$. We express the reduced problem using K'_{quad} in the regularizer and K'_{pred} in the prediction term, replacing 1_n with $1_{|\mathcal{Q}|}$, $y' = \Pi_{\mathcal{Q}}^{\text{Scaled}} y$ (which, by similar logic as in Section 3.1, preserves the objective function up to shifting by a constant), and $W' = \Pi_{\mathcal{Q}}^{\top} W \Pi_{\mathcal{Q}}$ since $v' = \Pi_{\mathcal{Q}}^{\top} v$. We note that while the reduced problem is a smaller convex program equivalent to the original, it is not necessarily expressible again in standard kernel form.

3.6.2 Kernel Binary Logistic Regression

With binary labels $y \in \{0, 1\}^n$, the optimization problem for kernel logistic regression is

$$\min F(\alpha, b) = -\left(\sum_{i=1}^{n} v_i \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right) \right) + \lambda \alpha^{\top} K \alpha,$$

where $\hat{y} = \sigma(K\alpha + b1_n)$ and σ is the sigmoid function.

Theorem 3.9. *A coloring $\mathcal{Q}$ is a reduction coloring for kernel binary logistic regression under the exact same conditions as for kernel ridge regression from Theorem 3.8.*

This will be proved in Appendix F.6. The reduced problem can be expressed in the same way as kernel ridge regression in Section 3.6.1, though y' will generally be fractional even though the original labels are binary.

To avoid fractional values in the ground truth, we can alternatively merge samples in each $T \in \mathcal{Q}$ into up to two reduced weighted samples, one with each label. The resulting reduced objective is given in Appendix G.

4 Experiments

We empirically evaluate our exact reduction framework on binary logistic regression, multiclass logistic regression, kernel binary logistic regression, ridge regression, and kernel ridge regression. The experiments evaluate the extent of achievable compression and whether the resulting reductions yield practical runtime improvements in training. As predicted by theory, all reduced problems yield solutions that are equivalent to those obtained from the original datasets.

We emphasize that exact reduction and approximate methods such as coresets or subsampling address fundamentally different objectives. Exact reduction produces a strictly equivalent optimization problem whereas approximate methods introduce controlled error in exchange for additional compression. As a result, these approaches are not directly comparable. Moreover, they are composable: exact reduction can be applied first to eliminate redundant structure, after which approximate methods may be applied to the reduced problem, potentially yielding improved efficiency-accuracy tradeoffs. Consequently, the most relevant comparison for our setting is the no-preprocessing setting rather than approximate compression techniques.

4.1 Datasets

We evaluate our method on binary logistic regression on the Titanic dataset; on multiclass logistic regression on MNIST; and on binary logistic regression, kernel binary logistic regression, ridge

regression, and kernel ridge regression on the UCI Adult dataset. All datasets used are publicly available and accessed via the OpenML Python API [Feurer et al., 2021], distributed under the CC-BY license.

For the Titanic dataset, the following preprocessing steps are applied:

- Categorical features are transformed using one-hot encoding, with the first category dropped to avoid redundancy.
- Missing categorical values are treated as separate categories.
- Missing numerical values are imputed using mean imputation.
- Identifying or high-cardinality features (name, ticket, cabin, boat, and body) are removed.

We also evaluate multiclass logistic regression on a binarized version of MNIST. For the MNIST experiments, we binarize all pixel values, which preserves the spatial sparsity patterns of the images while substantially increasing the frequency of exact redundancies and enabling nontrivial compression via color refinement.

For the Adult dataset, for kernel experiments, we subsample the dataset to 4,000 examples using a fixed random seed, RandomState(0), to keep kernel construction tractable. Categorical features are one-hot encoded, the resulting design matrix is binarized, and we use a linear kernel. We do not use subsampling for primal models. For classification experiments, the target is whether income exceeds \$50k. For regression experiments, we predict age from the remaining features.

4.2 Experimental Setup

All experiments were conducted on an Apple M3 Pro MacBook Pro with 12 CPU cores and 18 GB of memory. We used CPU-only implementations from scikit-learn and did not use GPU acceleration. Individual experimental runs on a single dataset typically required on the order of seconds to minutes of wall-clock time, depending on dataset size, and the full experimental evaluation required up to several hours of total compute over 50 trials. No substantial additional compute beyond the reported experiments was required for this work.

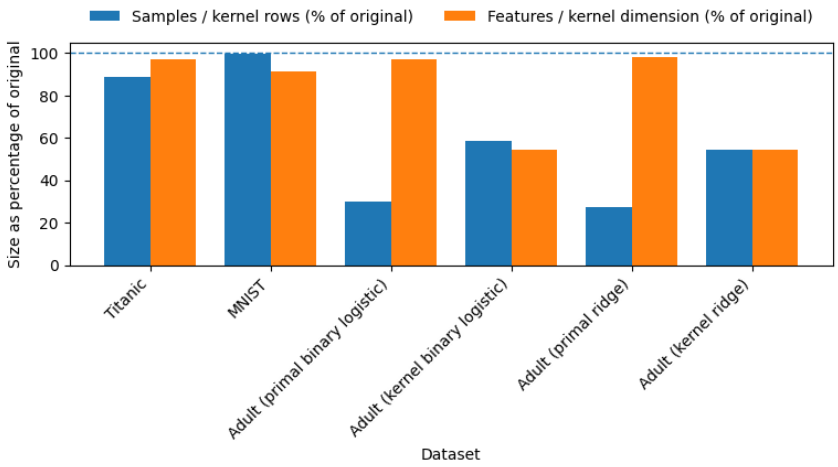
All reduction procedures are implemented in C++, while model training is performed using scikit-learn [Pedregosa et al., 2011]. Logistic regression models are unregularized, ridge and kernel ridge regression use a ridge coefficient of 10^{-6}, and kernel ridge uses the lsqr solver because the Adult linear kernel is rank-deficient so the default solver faces an ill-conditioned system. (For equivalence checks, kernel ridge uses the svd solver because lsqr is accurate only to about 10^{-4}; lsqr is used for timing runs.) We apply StandardScaler prior to training for the Titanic dataset and for primal binary logistic regression on the Adult dataset to improve numerical stability; the exact parameter transformation ensuring equivalence between reduced and original problems is described in Section 3.5.

For each dataset, we measure the number of samples and features retained after reduction and the total runtime, defined as the sum of reduction time and training time on the reduced dataset. To facilitate comparison across datasets, all runtimes are normalized by the training time on the original (unreduced) dataset. Predictive performance is identical up to solver tolerance and is therefore not reported separately.

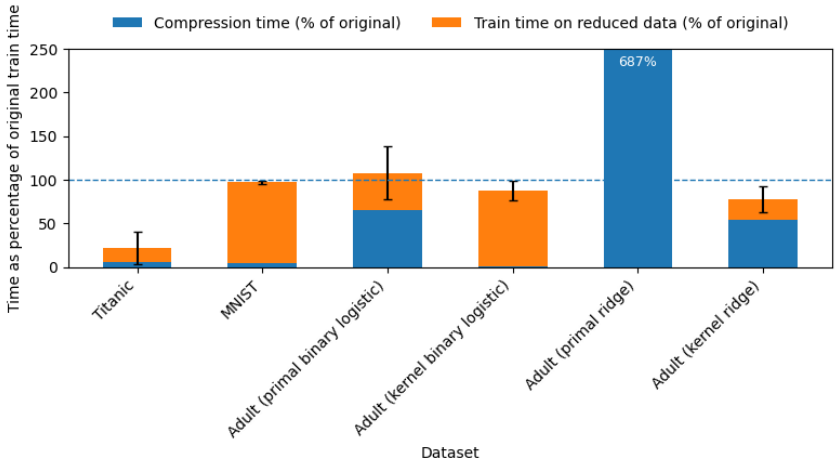
Our implementation uses exact floating-point equality for all refinement comparisons; the only approximation is that the initial coloring buckets columns by $\sum_i X_{ij} y_i$ rounded to 4 decimal places, used solely as an initialization heuristic before exact refinement. This did not affect any reported final partition (e.g. Titanic's partition is identical whether this seed rounds to 4, 8, or 12 decimals). Reported compression and reduction times measure color refinement given an already materialized data or kernel matrix; for kernel methods, the contribution reduces subsequent optimization cost but does not remove kernel materialization or storage costs.

4.3 Results

We examine the extent of sample and feature compression achieved by exact reduction in Figure 1a. Additionally, Figure 1b reports the total runtime with reduction, expressed as a percentage of the baseline training runtime on the original dataset. Error bars denote the sample standard deviation across 50 independent trials. For the total reduced runtime, we sum compression time and reduced



(a)



(b)

Figure 1: (a) Sample and feature compression percentage by dataset. (b) Runtime compression percentage by dataset.

training time, and approximate the standard deviation of the normalized runtime using first-order error propagation for the ratio between total reduced runtime and original training runtime, assuming independence between compression and reduced training times and using a linear approximation of the ratio.

These experiments illustrate that while exact reductions may be modest in low-redundancy settings such as MNIST, they remain effective in structured regimes, where measurable compression and runtime improvements can still be obtained.

5 Conclusion

We proposed an exact reduction framework for convex empirical risk minimization based on equitable partitions and color refinement. Our central result is general but conditional: it guarantees exactness given a valid reduction coloring, without claiming that such a coloring is automatically computable for an arbitrary convex objective. We instantiate it constructively for several model families, including linear and logistic regression, elastic-net regularization, and kernel methods. Experiments demonstrate that the proposed exact reductions can reduce problem size and improve end-to-end training time in structured settings, while explicit reduction constructions currently remain problem-specific; further limitations are discussed in Appendix B.

Acknowledgments and Disclosure of Funding

We would like to thank the MIT Program for Research in Mathematics, Engineering, and Science (PRIMES-USA) for making this project possible. The work of Z. Chen is supported in part by the National Science Foundation via grant DMS-2509011.

References

C. Berkholz, P. Bonsma, and M. Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. *Theory of Computing Systems*, 60:581–614, 2017. doi: 10.1007/s00224-016-9686-0.

Z. Chen, X. Chen, J. Liu, X. Wang, and W. Yin. Expressive power of graph neural networks for (mixed-integer) quadratic programs, 2025. Paper presented at the International Conference on Machine Learning, 13–19 July 2025.

P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020. doi: 10.1137/18M1209854.

M. Feurer, J. N. van Rijn, A. Kadra, P. Gijsbers, N. Mallik, S. Ravi, A. Mueller, J. Vanschoren, and F. Hutter. OpenML-Python: an extensible Python API for OpenML. *Journal of Machine Learning Research*, 22(100):1–5, 2021.

M. Grohe, K. Kersting, M. Mladenov, and E. Selman. Dimension reduction via colour refinement. *Algorithms - ESA*, 8737:505–516, 2014. doi: 10.1007/978-3-662-44777-2_42.

P. D. Loewen. Convex analysis with applications, 2001. Lecture notes, University of British Columbia. Available at `https://www.math.ubc.ca/~loew/m604/web-ho/convex.pdf`.

M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011. doi: 10.1561/2200000035.

M. Mladenov, L. Kleinhans, and K. Kersting. Lifted inference for convex quadratic programs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017. doi: 10.1609/aaai.v31i1.10841.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

M. Pilanci and M. J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, 17(53):1–38, 2016.

S. Ravanbakhsh, J. Schneider, and B. Poczos. Equivariance through parameter-sharing. In *International Conference on Machine Learning*, pages 2892–2901, 2017.

B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

A Prior Work

In this section, we review how equitable partitions lead to exact reductions for linear programs [Grohe et al., 2014] and convex quadratic programs [Mladenov et al., 2017].

A.1 Dimension Reduction for Linear Programs

We begin with linear programs (LPs), where color refinement-based dimension reduction is most transparent. Throughout, consider an LP of the form

$$\min\ c^\top x \quad \text{subject to } Ax \le b, \quad l \le x \le u.$$

We now state the conditions under which an LP admits an exact reduced formulation. Assume that

- for all $T \in \mathcal{Q}$, the triples (c_j, l_j, u_j) are identical for all $j \in T$;
- the pair $(\mathcal{P}, \mathcal{Q})$ is equitable with respect to A;
- for all $S \in \mathcal{P}$, the values b_i are identical for all $i \in S$.

Under these assumptions, define the reduced quantities

$$c' = \Pi_{\mathcal{Q}}^\top c, \qquad A' = \Pi_{\mathcal{P}}^{\text{Scaled}} A \Pi_Q, \qquad b' = \Pi_{\mathcal{P}}^{\text{Scaled}} b,$$
$$x' = \Pi_{\mathcal{Q}}^{\text{Scaled}} x, \qquad l' = \Pi_{\mathcal{Q}}^{\text{Scaled}} l, \qquad u' = \Pi_{\mathcal{Q}}^{\text{Scaled}} u.$$

The reduced LP is obtained by replacing (c, A, b, l, u) with (c', A', b', l', u'). It is established in Grohe et al. [2014] that x is an optimal solution of the original LP if and only if x' is an optimal solution of the reduced LP.

A.2 Dimension Reduction for Quadratic Programs

We next consider convex quadratic programs (QPs) of the form

$$\min\ \tfrac{1}{2} x^\top Q x + c^\top x \quad \text{subject to } Ax \le b, \quad l \le x \le u,$$

where $Q \succeq 0$ and is symmetric. The reduction procedure largely mirrors that of linear programs, with one additional requirement ensuring compatibility with the quadratic term.

In addition to the assumptions in the LP case, we require that the partition $(\mathcal{Q}, \mathcal{Q})$ be equitable with respect to the matrix Q. (Here, $\mathcal{Q}$ partitions both the rows and columns of Q.) Under this condition, the reduced quadratic matrix is given by

$$Q' = \Pi_{\mathcal{Q}}^\top Q \Pi_{\mathcal{Q}}.$$

As shown in Mladenov et al. [2017], this construction preserves convexity and optimality: solutions of the reduced QP correspond exactly to solutions of the original problem.

B Limitations

The effectiveness of the proposed reduction framework depends strongly on the presence of structural redundancy in the data. In settings where features and samples are fully continuous and lack repeated patterns, the coarsest equitable partition may be close to the identity, yielding little or no compression. As a result, the practical benefits of the method are most pronounced in structured regimes, such as sparse, categorical, quantized, or repeated-pattern data, and may be limited in dense real-valued settings.

Our approach also relies on explicit knowledge of the optimization problem structure. While we provide constructive conditions for several common convex ERM models, extending these reductions to more general objectives may require additional analysis. In particular, the applicability of the framework is currently restricted to convex programs with sufficient differentiability structure, and does not directly extend to non-convex or non-differentiable objectives without further modifications.

From a computational perspective, the reduction procedure itself incurs overhead. Although color refinement is efficient in practice and often outweighed by the savings from solving a smaller

optimization problem, this tradeoff depends on the problem size and the degree of achievable compression.

Finally, our empirical evaluation is limited to a small set of benchmark datasets and emphasizes structured representations (e.g. binarized or kernel-induced features) where exact redundancy is more likely to occur.

To empirically examine the limitations of our method, we present a controlled study in Appendix I.

C Proof of Reduction Theorem

In this proof, we will occasionally use color-based indexing in cases where all elements of a color are known to be equal. For example, we might say x_T for $T \in \mathcal{Q}$ if it is known that x_j is equal for all $j \in T$.

Proof of Theorem 2.3. Suppose that $(\mathcal{P}, \mathcal{Q})$ is a reduction coloring. Let $\hat{x} = \Pi_{\mathcal{Q}} \Pi_{\mathcal{Q}}^{\text{Scaled}} x$. Note that if $j_1 \in T$ and $T \in \mathcal{Q}$, then

$$\hat{x}_{j_1} = \frac{\sum_{j_2 \in T} x_{j_2}}{|T|}.$$

We prove our reduction theorem in three steps:

1. The objective function does not increase when x is changed to $\hat{x}$.
2. If x is feasible, $\hat{x}$ must be too.
3. The above statements imply equivalence of the original and reduced problems.

Firstly, we show that $F(\hat{x}) \leq F(x)$ using the Lagrangian, inspired by Chen et al. [2025]. Suppose that x is fixed, and we wish to minimize $F(z)$ such that

$$\sum_{j \in T} z_j = \sum_{j \in T} x_j$$

for all $T \in \mathcal{Q}$. It suffices to show that $z = \hat{x}$ is a minimum, and hence, since the objective is convex and it is obvious that $z = \hat{x}$ satisfies our condition, that the partial derivative of the Lagrangian

$$L = F(z) - \sum_{T \in \mathcal{Q}} \lambda_T \left(\left(\sum_{j \in T} z_j \right) - \sum_{j \in T} x_j \right)$$

with respect to each z_{j_s} is 0 if $z = \hat{x}$. We have that

$$\frac{\partial L}{\partial z_j} = \frac{\partial F}{\partial z_j} - \lambda_T$$

if $j \in T$. Then our claim is true if we have equal $\left.\frac{\partial F}{\partial z_j}\right|_{z=\hat{x}}$ for all j in the same color T, which is one of our reduction coloring conditions.

Secondly, we show that $\hat{x}$ is feasible. Due to our requirement that each $T \in \mathcal{Q}$ has the same (l_j, u_j) for all $j \in T$, we immediately have that $\hat{x}$ is in the box domain. Additionally, we know that

$$\sum_{i \in S} G_i(\hat{x}) \leq \sum_{i \in S} G_i(x) \leq |S| b_S.$$

The first inequality follows by applying the same equality-constrained convex minimization argument to the convex function $\sum_{i \in S} G_i$ using the gradient condition from the reduction coloring for $\sum_{i \in S} G_i$. The second inequality is true by assumption and due to our requirement that b_i is equal for all $i \in S$. In fact, $G_i(\hat{x}) \leq b_i$ for all individual $1 \leq i \leq m$ because we require that $G_i(\hat{x})$ is equal for all $i \in S$.

Finally, we show that the above implies optimality of x' on the reduced problem. Since F' is given simply by replacing each x_j in F with x_T where $j \in T$, we have that

$$F'(x') = F(\hat{x}),$$

so if x' is not optimal, then $\hat{x}$ and hence x is not optimal, which contradicts our assumption. We can use similar logic with the reduced constraints: $G'(x') = G(\hat{x})$, so x' is feasible too.

For the reverse direction, note that $\hat{x} = \Pi_{\mathcal{Q}} x'$. Hence, if x' is optimal, it is impossible for $\hat{x}$ to not be optimal. Otherwise, there exists some feasible y with $F(y) < F(\hat{x}) \implies F'(y') < F'(x')$, contradiction. □

Remark C.1. To see why the reduced objective and constraints are convex, note that every reduced expression is given by a scaled intersection of the original curve and the affine space given by setting all x_j equal for all $j \in T$ and any fixed $T \in \mathcal{Q}$. If a function is convex over a convex set, then it is also convex over any affine subspace of this set [Loewen, 2001], showing that the reduced expressions are convex.

D Proof and Example of Reduction Coloring Coarseness Compared to Permutation Invariance

Proof of Theorem 2.4. Consider any two variables j_1, j_2 in the same orbit under Γ. Then there exists some $(\pi, \sigma) \in \Gamma$ such that $\pi(j_1) = j_2$. Let $\hat{x}$ be any point in the box domain that is constant within each $T \in \mathcal{Q}_\Gamma$, meaning $\pi(\hat{x}) = \hat{x}$.

Since $F(\pi(x)) = F(x)$ for all x in the box domain, viewing π as the permutation matrix P_π (so $\pi(x) = P_\pi x$) and differentiating gives, by the chain rule,

$$P_\pi^\top \nabla F(P_\pi x) = \nabla F(x)$$

for all such x. Evaluating at $x = \hat{x}$ and using $P_\pi \hat{x} = \hat{x}$ yields

$$P_\pi^\top \nabla F(\hat{x}) = \nabla F(\hat{x}),$$

which is equivalent to $\nabla F(\hat{x}) = P_\pi \nabla F(\hat{x})$ since $P_\pi^{-1} = P_\pi^\top$ for any π. Therefore, $\big(\nabla F(\hat{x})\big)_j = \big(\nabla F(\hat{x})\big)_{\pi(j)}$ for all j, so

$$\left.\frac{\partial F}{\partial x_{j_1}}\right|_{x=\hat{x}} = \left.\frac{\partial F}{\partial x_{j_2}}\right|_{x=\hat{x}}.$$

For constraints, fix any constraint orbit $S \in \mathcal{P}_\Gamma$. We claim that $\sum_{i\in S} G_i(\pi(x)) = \sum_{i\in S} G_i(x)$ for all x in the box domain. By the definition of Γ, we have $G_{\sigma(i)}(\pi(x)) = G_i(x)$ for some permutation σ for all $1 \le i \le m$, so summing over $i \in S$ gives

$$\sum_{i\in S} G_{\sigma(i)}(\pi(x)) = \sum_{i\in S} G_i(x).$$

Since S is an orbit, it is Γ-invariant. Hence, $\sigma(S) = S$, and σ restricts to a bijection on S. Thus, $\sum_{i\in S} G_{\sigma(i)}(\pi(x))$ is simply a reindexing:

$$\sum_{i\in S} G_{\sigma(i)}(\pi(x)) = \sum_{i\in S} G_i(\pi(x)) = \sum_{i\in S} G_i(x).$$

Using the same logic as above, this implies that

$$\left.\frac{\partial}{\partial x_{j_1}}\left(\sum_{i\in S} G_i\right)\right|_{x=\hat{x}} = \left.\frac{\partial}{\partial x_{j_2}}\left(\sum_{i\in S} G_i\right)\right|_{x=\hat{x}}.$$

Finally, by definition of Γ, constraints in the same orbit satisfy $b_{\sigma(i)} = b_i$ and $G_{\sigma(i)}(\pi(x)) = G_i(x)$. Evaluating at $x = \hat{x}$ with $\pi(\hat{x}) = \hat{x}$ implies that $G_i(\hat{x})$ and b_i are both constant within each $S \in \mathcal{P}_\Gamma$. Moreover, $\pi(l) = l$ and $\pi(u) = u$ imply that (l_j, u_j) is constant within each $T \in \mathcal{Q}_\Gamma$. Therefore, $(\mathcal{P}_\Gamma, \mathcal{Q}_\Gamma)$ is a reduction coloring, so the coarsest reduction coloring $(\mathcal{P}, \mathcal{Q})$ must be at least as coarse as $(\mathcal{P}_\Gamma, \mathcal{Q}_\Gamma)$. □

Now, we provide an example where Theorem 2.3 can be used for compression even when no nontrivial permutation symmetry exists. Consider an unregularized linear regression problem on the following

data:

$$X = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 4 & 2 & 2 \\ 2 & 4 & 2 \\ 3 & 3 & 1 \end{bmatrix}, \qquad y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 7 \end{bmatrix}.$$

There is clearly no pair of permutations (π, σ) that preserves the data. For example, swapping the first and second features sends the sample $(3, 1, 0)$ with label 0 to $(1, 3, 0)$, which occurs only with label 1, so no sample permutation can restore equality. However, $\mathcal{P} = \{\{1, 2\}, \{3, 4\}, \{5\}\}$ and $\mathcal{Q} = \{\{1, 2\}, \{3\}\}$ is a reduction coloring. Thus, the problem is equivalent to one with the following compressed data:

$$X' = \begin{bmatrix} 4 & 0 \\ 6 & 2 \\ 6 & 1 \end{bmatrix}, \qquad y' = \begin{bmatrix} 0.5 \\ 0.5 \\ 7 \end{bmatrix},$$

with

$$W' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We have that $w = \begin{bmatrix} 6.5 \\ 6.5 \\ -6.5 \end{bmatrix}$, $b = b' = -25.5$, and $w' = \begin{bmatrix} 6.5 \\ -6.5 \end{bmatrix}$, as expected.

E Proof of Time Complexity of Algorithm 1

Proof. The first time the color R of some index $1 \le i \le m$ or $1 \le j \le n$ is pushed into S_{refine}, it has size $|R|$. If the color of i or j ever gets pushed into S_{refine} again, then at some point, R must have been split, with i or j being assigned to some color in $\mathcal{S} \setminus \{\gamma\}$. Hence, the current color of i or j has size at most $\frac{|R|}{2}$. This implies that each i can be pushed into S_{refine} at most $\log_2(m)$ times, and each j can be pushed at most $\log_2(n)$ times. Therefore, $\sum_{R \in \mathcal{Q} \text{ popped}} |R| = O(n \log(n))$, and $\sum_{R \in \mathcal{P} \text{ popped}} |R| = O(m \log(m))$.

When we pop a color $R \in \mathcal{P}$ and process it, we must examine all entries of A whose first index lies in R, which is $|R|n$ entries. Hence, the total work over all colors is

$$O\left(n \sum_{R \in \mathcal{P} \text{ popped}} |R|\right) = O(mn \log(m)).$$

Similarly, the total work over all colors in $\mathcal{Q}$ is $O(mn \log(n))$. Combining these, we get the desired time complexity of $O(mn(\log(m) + \log(n)))$. □

F Proofs and Examples for Section 3

F.1 Proof of Reduction for Linear Regression and Preservation of Closed-Form

Proof of Theorem 3.1. At some point $\hat{w}$ with $\hat{w}_{j_1} = \hat{w}_{j_2}$ if j_1 and j_2 share a color in $\mathcal{Q}$,

$$\begin{aligned} \left.\frac{\partial F}{\partial w_j}\right|_{w=\hat{w}} &= 2 \sum_{i=1}^{n} X_{ij}(\hat{y}_i - y_i) \\ &= 2 \sum_{i=1}^{n} X_{ij} \left(\left(b + \sum_{k=1}^{D} X_{ik} w_k \right) - y_i \right) \\ &= 2 \sum_{i=1}^{n} X_{ij} \left(\left(b + \sum_{T \in \mathcal{Q}} w_T \sum_{k \in T} X_{ik} \right) - y_i \right), \end{aligned}$$

where we use w_T to denote the constant value of w_j for all $j \in T$.

Under our reduction conditions, for any $T \in \mathcal{Q}$ and $j_1, j_2 \in T$,

$$\begin{aligned}
&\left(\left.\frac{\partial F}{\partial w_{j_1}}\right|_{w=\hat{w}}\right) - \left(\left.\frac{\partial F}{\partial w_{j_2}}\right|_{w=\hat{w}}\right) \\
&\quad = 2\sum_{i=1}^{n}(X_{ij_1} - X_{ij_2})\left(\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) - y_i\right) \\
&\quad = 2\sum_{i=1}^{n}(X_{ij_1} - X_{ij_2})\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) - \left(\sum_{i=1}^{n} X_{ij_1}y_i - X_{ij_2}y_i\right) \\
&\quad = 2\sum_{i=1}^{n}(X_{ij_1} - X_{ij_2})\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right).
\end{aligned}$$

Due to equitability of $(\mathcal{P}, \mathcal{Q})$ on X, we have constant $\sum_{j\in T} X_{ij}$ for all $i \in S$ and constant $\sum_{i\in S} X_{ij}$ for all $j \in T$ if $S \in \mathcal{P}$ and $T \in \mathcal{Q}$. Hence, the prediction for each sample,

$$\hat{y}_i = b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij},$$

is equal for all i that share a color $S \in \mathcal{P}$, and we can denote this value as $\hat{y}_S$. Using this, we can further simplify:

$$\begin{aligned}
\sum_{i=1}^{n}(X_{ij_1} - X_{ij_2})\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) &= \sum_{S\in\mathcal{P}} \hat{y}_S \left(\left(\sum_{i\in S} X_{ij_1}\right) - \sum_{i\in S} X_{ij_2}\right) \\
&= \sum_{S\in\mathcal{P}} \hat{y}_S \cdot 0 \\
&= 0.
\end{aligned}$$

Hence, $\left.\frac{\partial F}{\partial w_{j_1}}\right|_{w=\hat{w}} = \left.\frac{\partial F}{\partial w_{j_2}}\right|_{w=\hat{w}}$ for all $j_1, j_2 \in T$. □

We can further verify that the closed-form optimal w,

$$w = \left(X^\top W X\right)^{-1} X^\top W y,$$

is preserved. (Note that we assume $W = I$ in the original problem. Additionally, we assume X has been augmented with a column of ones so that w includes a bias term.) In the reduced problem, this formula becomes

$$\begin{aligned}
w' &= \left((X')^\top W' X'\right)^{-1} (X')^\top W' \Pi_{\mathcal{P}}^{\text{Scaled}} y \\
&= \left(\Pi_{\mathcal{Q}}^\top X^\top \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top W' \Pi_{\mathcal{P}}^{\text{Scaled}} X \Pi_{\mathcal{Q}}\right)^{-1} \Pi_{\mathcal{Q}}^\top X^\top \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top W' \Pi_{\mathcal{P}}^{\text{Scaled}} y,
\end{aligned}$$

which implies that

$$\Pi_{\mathcal{Q}}^\top X^\top \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} X \Pi_{\mathcal{Q}} w' = \Pi_{\mathcal{Q}}^\top X^\top \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y.$$

Since $\Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} \in \mathbb{R}^{n\times n}$ satisfies $\left(\Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}}\right)_{i_1 i_2} = \frac{1}{|S|}$ for any $i_1, i_2 \in S$, left-multiplication by $\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}}$ averages the entries within each color $S \in \mathcal{P}$. Additionally, if $i_1, i_2 \in S$, then we have that

$$\begin{aligned}
(X\hat{w})_{i_1} &= \sum_{T\in\mathcal{Q}} w'_T \sum_{j\in T} X_{i_1 j} \\
&= \sum_{T\in\mathcal{Q}} w'_T \sum_{j\in T} X_{i_2 j} \\
&= (X\hat{w})_{i_2}
\end{aligned}$$

from equitability. Thus,

$$\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} X\hat{w} = X\hat{w}, \tag{1}$$

so we can simplify:

$$\Pi_{\mathcal{Q}}^{\top} X^{\top} X \hat{w} = \Pi_{\mathcal{Q}}^{\top} X^{\top} \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y.$$

In other words, for any $T \in \mathcal{Q}$, we have that

$$\sum_{j \in T} \left(X^{\top} X \hat{w}\right)_j = \sum_{j \in T} \left(X^{\top} \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_j.$$

We can rewrite $\left(X^{\top} X \hat{w}\right)_j$ as follows:

$$\begin{aligned}
\left(X^{\top} X \hat{w}\right)_j &= \sum_{k=1}^{D} \left(X^{\top} X\right)_{jk} \hat{w}_k \\
&= \sum_{k=1}^{D} \hat{w}_k X_k^{\top} X_j \\
&= \left(\sum_{T \in \mathcal{Q}} w'_T \sum_{k \in T} X_k\right)^{\top} X_j \\
&= \sum_{i=1}^{n} \left(\sum_{T \in \mathcal{Q}} w'_T \sum_{k \in T} X_{ik}\right) X_{ij},
\end{aligned}$$

where X_j is the column vector representing the jth feature. We have that

$$\begin{aligned}
\sum_{i=1}^{n} \left(\sum_{T \in \mathcal{Q}} w'_T \sum_{k \in T} X_{ik}\right) X_{ij} &= \sum_{S \in \mathcal{P}} \sum_{T \in \mathcal{Q}} w'_T \sum_{i \in S} X_{ij} \sum_{k \in T} X_{ik} \\
&= \sum_{S \in \mathcal{P}} \sum_{T \in \mathcal{Q}} w'_T X'_{ST} \sum_{i \in S} X_{ij}
\end{aligned}$$

from equitability of $(\mathcal{P}, \mathcal{Q})$ on X. It also follows from equitability that $\sum_{i \in S} X_{ij_1} = \sum_{i \in S} X_{ij_2}$ if j_1 and j_2 share a color in $\mathcal{Q}$, so

$$\sum_{S \in \mathcal{P}} \sum_{T \in \mathcal{Q}} w'_T X'_{ST} \sum_{i \in S} X_{ij} = \left(X^{\top} X \hat{w}\right)_j$$

is constant across all $j \in T$. Recall that

$$\sum_{j \in T} \left(X^{\top} X \hat{w}\right)_j = \sum_{j \in T} \left(X^{\top} \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_j$$

for any $T \in \mathcal{Q}$. Since left-multiplication by $\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}}$ averages each color $S \in \mathcal{P}$, $\left(\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_i$ is equal for all $i \in S$, and we can express it as $\left(\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_S$. This means that

$$\left(X^{\top} \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_j = \sum_{S \in \mathcal{P}} \left(\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_S \sum_{i \in S} X_{ij}$$

is also constant across $j \in T$ due to equitability. These show that

$$X^{\top} X \hat{w} = X^{\top} \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^{\top} \Pi_{\mathcal{P}}^{\top} \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y. \tag{2}$$

We also have for any $T \in \mathcal{Q}$ that

$$\begin{aligned}
\sum_{j\in T}\left(X^\top\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_j &= \sum_{j\in T}\sum_{S\in\mathcal{P}} y'_S \sum_{i\in S} X_{ij} \\
&= \sum_{S\in\mathcal{P}} y'_S \sum_{i\in S}\sum_{j\in T} X_{ij} \\
&= \sum_{S\in\mathcal{P}} |S|\, y'_S\, X_{ST} \\
&= \sum_{S\in\mathcal{P}}\sum_{i\in S} y_i X_{ST} \\
&= \sum_{S\in\mathcal{P}}\sum_{i\in S} y_i \sum_{j\in T} X_{ij} \\
&= \sum_{i=1}^{n} y_i \sum_{j\in T} X_{ij} \\
&= \sum_{j\in T}\left(X^\top y\right)_j,
\end{aligned}$$

where $X_{ST} = \sum_{j\in T} X_{ij}$ is constant regardless of our choice of $i \in S$ due to equitability. Since we require that $\left(X^\top y\right)_j$ is equal across $j \in T$ and

$$\left(X^\top\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y\right)_j = \sum_{S\in\mathcal{P}} y'_S \sum_{i\in S} X_{ij}$$

is also equal across $j \in T$ due to equitability, we have that

$$X^\top\left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top \Pi_{\mathcal{P}}^\top \Pi_{\mathcal{P}} \Pi_{\mathcal{P}}^{\text{Scaled}} y = X^\top y, \tag{3}$$

so Equation 2 becomes

$$X^\top X\hat{w} = X^\top y \implies \hat{w} = \left(X^\top X\right)^{-1} X^\top y.$$

F.2 Proof of Reduction for Binary Logistic Regression

Proof of Theorem 3.4. If $\hat{w}_{j_1} = \hat{w}_{j_2}$ for any j_1 and j_2 that share a color in $\mathcal{Q}$,

$$\begin{aligned}
\left.\frac{\partial F}{\partial w_j}\right|_{w=\hat{w}} &= \sum_{i=1}^{n} v_i X_{ij}\left(\hat{y}_i - y_i\right) \\
&= \sum_{i=1}^{n} v_i X_{ij}\left(\sigma\left(b + \sum_{k=1}^{D} X_{ik} w_k\right) - y_i\right) \\
&= \sum_{i=1}^{n} v_i X_{ij}\left(\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) - y_i\right).
\end{aligned}$$

Under our reduction conditions, for any $T \in \mathcal{Q}$ and $j_1, j_2 \in T$,

$$\begin{aligned}
&\left(\frac{\partial F}{\partial w_{j_1}}\bigg|_{w=\hat{w}}\right) - \left(\frac{\partial F}{\partial w_{j_2}}\bigg|_{w=\hat{w}}\right) \\
&\quad= \sum_{i=1}^{n} v_i (X_{ij_1} - X_{ij_2}) \left(\sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{k\in T} X_{ik}\right) - y_i\right) \\
&\quad= \left(\sum_{i=1}^{n} v_i (X_{ij_1} - X_{ij_2})\, \sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij}\right)\right) - \sum_{i=1}^{n} v_i \left(X_{ij_1} y_i - X_{ij_2} y_i\right) \\
&\quad= \sum_{i=1}^{n} v_i (X_{ij_1} - X_{ij_2})\, \sigma\left(b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij}\right) \\
&\quad= \sum_{S\in\mathcal{P}} \frac{v'_S}{|S|}\, \sigma(z_S) \left(\left(\sum_{i\in S} X_{ij_1}\right) - \sum_{i\in S} X_{ij_2}\right) \\
&\quad= \sum_{S\in\mathcal{P}} \frac{v'_S}{|S|}\, \sigma(z_S) \cdot 0 \\
&\quad= 0,
\end{aligned}$$

where z_S represents the value of the logit

$$z_i = b + \sum_{T\in\mathcal{Q}} w_T \sum_{j\in T} X_{ij},$$

which is equal for all $i \in S$ for some $S \in \mathcal{P}$. Hence, $\frac{\partial F}{\partial w_{j_1}}\Big|_{w=\hat{w}} = \frac{\partial F}{\partial w_{j_2}}\Big|_{w=\hat{w}}$ for all $j_1, j_2 \in T$. $\square$

F.3 Proof of Reduction for Multiclass Logistic Regression

Proof of Theorem 3.5. We have that

$$\begin{aligned}
\frac{\partial F}{\partial W_{jc}} &= \sum_{i=1}^{n} v_i X_{ij} (\hat{y}_{ic} - 1\{y_i = c\}) \\
&= \sum_{i=1}^{n} v_i X_{ij} \left(\frac{e^{b_c + \sum_{k=1}^{D} X_{ik} W_{kc}}}{\sum_{l=1}^{K} e^{b_l + \sum_{k=1}^{D} X_{ik} W_{kl}}} - 1\{y_i = c\}\right).
\end{aligned}$$

If $(\mathcal{P}, \mathcal{Q})$ is a reduction coloring and the two rows $\hat{W}_{j_1} = \hat{W}_{j_2}$ whenever j_1 and j_2 share a color in $\mathcal{Q}$ (the entries within $\hat{W}_{j_1}$ and $\hat{W}_{j_2}$ do not necessarily have to be equal), then

$$\begin{aligned}
\frac{\partial F}{\partial W_{jc}}\bigg|_{W=\hat{W}} &= \sum_{i=1}^{n} v_i X_{ij} \left(\frac{e^{b_c + \sum_{T\in\mathcal{Q}} \sum_{k\in T} X_{ik} W_{kc}}}{\sum_{l=1}^{K} e^{b_l + \sum_{T\in\mathcal{Q}} \sum_{k\in T} X_{ik} W_{kl}}} - 1\{y_i = c\}\right) \\
&= \sum_{i=1}^{n} v_i X_{ij} \left(\frac{e^{b_c + \sum_{T\in\mathcal{Q}} W_{Tc} \sum_{k\in T} X_{ik}}}{\sum_{l=1}^{K} e^{b_l + \sum_{T\in\mathcal{Q}} W_{Tl} \sum_{k\in T} X_{ik}}} - 1\{y_i = c\}\right) \\
&= \sum_{S\in\mathcal{P}} \sum_{i\in S} v_i X_{ij} \left(\hat{y}_{Sc} - 1\{y_i = c\}\right),
\end{aligned}$$

where

$$\hat{y}_{Sc} = \frac{e^{b_c + \sum_{T\in\mathcal{Q}} W_{Tc} \sum_{k\in T} X_{ik}}}{\sum_{l=1}^{K} e^{b_l + \sum_{T\in\mathcal{Q}} W_{Tl} \sum_{k\in T} X_{ik}}}$$

is constant across $i \in S$ if $(\mathcal{P}, \mathcal{Q})$ is equitable on X.

Then under reduction conditions, for any $T \in \mathcal{Q}$ and $j_1, j_2 \in T$,

$$\begin{aligned}
&\left(\frac{\partial F}{\partial W_{j_1 c}}\bigg|_{W=\hat{W}}\right) - \left(\frac{\partial F}{\partial W_{j_2 c}}\bigg|_{W=\hat{W}}\right) \\
&\quad= \sum_{S\in\mathcal{P}} \sum_{i\in S} v_i (X_{ij_1} - X_{ij_2}) \left(\hat{y}_{Sc} - 1\{y_i = c\}\right) \\
&\quad= \left(\sum_{S\in\mathcal{P}} \frac{v'_S}{|S|} \hat{y}_{Sc} \sum_{i\in S} (X_{ij_1} - X_{ij_2})\right) - \sum_{i=1}^{n} v_i (X_{ij_1} - X_{ij_2}) \cdot 1\{y_i = c\} \\
&\quad= \sum_{S\in\mathcal{P}} \frac{v'_S}{|S|} \hat{y}_{Sc} \left(\left(\sum_{i\in S} X_{ij_1}\right) - \sum_{i\in S} X_{ij_2}\right) \\
&\quad= 0.
\end{aligned}$$

Hence, $\frac{\partial F}{\partial W_{j_1 c}}\Big|_{W=\hat{W}} = \frac{\partial F}{\partial W_{j_2 c}}\Big|_{W=\hat{W}}$ for all $j_1, j_2 \in T$. □

F.4 Preservation of Ridge Regression Closed-Form

We can verify that the closed-form solution for ridge regression,

$$w = \left(X^\top W X + \lambda_2 I\right)^{-1} X^\top W y,$$

is preserved by our reduction coloring for linear regression. Note that the regularization term in the reduced problem is

$$\sum_{T\in\mathcal{Q}} \lambda_2 |T| (w'_T)^2.$$

Hence, in the reduced problem, we have

$$\left(\Pi_{\mathcal{Q}}^\top X^\top \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top W' \Pi_{\mathcal{P}}^{\text{Scaled}} X \Pi_{\mathcal{Q}} + Q'\right) w' = \Pi_{\mathcal{Q}}^\top X^\top \left(\Pi_{\mathcal{P}}^{\text{Scaled}}\right)^\top W' \Pi_{\mathcal{P}}^{\text{Scaled}} y,$$

where Q' is a diagonal matrix given by $Q'_{TT} = \lambda_2 |T|$ for all $T \in \mathcal{Q}$. Because of Equation 1 and Equation 3 in Appendix F.1, we can rewrite this as

$$\Pi_{\mathcal{Q}}^\top X^\top X \hat{w} + Q' w' = \Pi_{\mathcal{Q}}^\top X^\top y.$$

This implies that

$$\lambda_2 |T| w'_T + \sum_{j\in T} \left(X^\top X \hat{w}\right)_j = \sum_{j\in T} \left(X^\top y\right)_j$$

for any $T \in \mathcal{Q}$. But since $\left(X^\top X \hat{w}\right)_j$ and $\left(X^\top y\right)_j$ are both constant across $j \in T$, dividing by $|T|$ gives

$$\lambda_2 \hat{w} + X^\top X \hat{w} = X^\top y \implies \hat{w} = \left(X^\top X + \lambda_2 I\right)^{-1} X^\top y.$$

F.5 Proof of Reduction for Kernel Ridge Regression

Proof of Theorem 3.8. We have that

$$\frac{\partial F}{\partial \alpha_j}\bigg|_{\alpha=\hat{\alpha}} = 2\left(\sum_{i=1}^{n} v_i K_{ij} (K\hat{\alpha} + b 1_n - y)_i\right) + 2\lambda (K\hat{\alpha})_j.$$

Suppose that $T_1 \in \mathcal{Q}$ and $j_1, j_2 \in T_1$. From equitability on K,

$$\begin{aligned}
(K\hat{\alpha})_{j_1} &= \sum_{i=1}^{n} K_{ij_1} \hat{\alpha}_i \\
&= \sum_{T_2\in\mathcal{Q}} \hat{\alpha}_{T_2} \sum_{i\in T_2} K_{ij_1} \\
&= \sum_{T_2\in\mathcal{Q}} \hat{\alpha}_{T_2} \sum_{i\in T_2} K_{ij_2} \\
&= (K\hat{\alpha})_{j_2},
\end{aligned}$$

so the $2\lambda(K\hat{\alpha})_j$ term is constant across all j in the same color. Also, since v_i is constant on each $T \in \mathcal{Q}$, we can write $v_i = \frac{v'_T}{|T|}$ for all $i \in T$. Hence, we can decompose the summation term of the derivative with respect to α_{j_1} as

$$\begin{aligned}
&\left(\sum_{i=1}^{n} v_i K_{ij_1}(K\hat{\alpha})_i\right) + \left(\sum_{i=1}^{n} v_i K_{ij_1} b\right) - \sum_{i=1}^{n} v_i K_{ij_1} y_i \\
&\quad= \left(\sum_{T\in\mathcal{Q}} \frac{v'_T}{|T|}(K\hat{\alpha})_T \sum_{i\in T} K_{ij_1}\right) + b\left(\sum_{T\in\mathcal{Q}} \frac{v'_T}{|T|}\sum_{i\in T} K_{ij_1}\right) - \sum_{i=1}^{n} v_i K_{ij_1} y_i \\
&\quad= \left(\sum_{T\in\mathcal{Q}} \frac{v'_T}{|T|}(K\hat{\alpha})_T \sum_{i\in T} K_{ij_2}\right) + b\left(\sum_{T\in\mathcal{Q}} \frac{v'_T}{|T|}\sum_{i\in T} K_{ij_2}\right) - \sum_{i=1}^{n} v_i K_{ij_2} y_i \\
&\quad= \left(\sum_{i=1}^{n} v_i K_{ij_2}(K\hat{\alpha})_i\right) + \left(\sum_{i=1}^{n} v_i K_{ij_2} b\right) - \sum_{i=1}^{n} v_i K_{ij_2} y_i,
\end{aligned}$$

which proves Theorem 3.8. □

F.6 Proof of Reduction for Kernel Binary Logistic Regression

Proof of Theorem 3.9. The derivative of the loss function is

$$\begin{aligned}
\left.\frac{\partial F}{\partial \alpha_j}\right|_{\alpha=\hat{\alpha}} &= \left(\sum_{i=1}^{n} v_i K_{ij}(\hat{y}_i - y_i)\right) + 2\lambda(K\hat{\alpha})_j \\
&= \left(\sum_{i=1}^{n} v_i K_{ij}\,\sigma((K\hat{\alpha})_i + b)\right) - \left(\sum_{i=1}^{n} v_i K_{ij} y_i\right) + 2\lambda(K\hat{\alpha})_j \\
&= \left(\sum_{T\in\mathcal{Q}} \frac{v'_T}{|T|}\sigma((K\hat{\alpha})_T + b)\sum_{i\in T} K_{ij}\right) - \left(\sum_{i=1}^{n} v_i K_{ij} y_i\right) + 2\lambda(K\hat{\alpha})_j,
\end{aligned}$$

where the last equality is true because we know from the proof of Theorem 3.8 in Appendix F.5 that $(K\hat{\alpha})_j$ is constant across $j \in T$ for any $T \in \mathcal{Q}$. This derivative as a whole is constant across $j \in T$ because we have from equitability that the first term in the above expression is constant, and from Appendix F.5 that the latter two terms are also constant. □

G Reduced Objective for Kernel Binary Logistic Regression

Each $T \in \mathcal{Q}$ is split into up to two subsets $T^{(0)} = \{i \in T \mid y_i = 0\}$ and $T^{(1)} = \{i \in T \mid y_i = 1\}$. The resulting partition is denoted by $\mathcal{P}$. As in Section 3.6.1, the regularizer uses $K'_{\text{quad}} = \Pi_{\mathcal{Q}}^{\top} K \Pi_{\mathcal{Q}}$, but here the prediction term uses $K'_{\text{logit}} = \Pi_{\mathcal{P}}^{\text{Scaled}} K \Pi_{\mathcal{Q}}$, which is generally rectangular since $\mathcal{P}$ refines $\mathcal{Q}$, rather than square as in the regression case. Hence, the reduced objective can be written as follows:

$$\min\ F'(\alpha', b') = -\left(\sum_{S\in\mathcal{P}} v'_S\,(y'_S \log \hat{y}'_S + (1 - y'_S)\log(1 - \hat{y}'_S))\right) + \lambda \alpha'^{\top} K'_{\text{quad}} \alpha',$$

where $v' = \Pi_{\mathcal{P}}^{\top} v$, $y' = \Pi_{\mathcal{P}}^{\text{Scaled}} y$, and $\hat{y}' = \sigma(K'_{\text{logit}}\alpha' + b1_{|\mathcal{P}|})$.

H Full Experimental Results

We used a binary logistic regression model with regularization turned off (no penalty term) and a maximum of 100,000 iterations. All remaining hyperparameters were set to their default values in scikit-learn version 1.6.1 [Pedregosa et al., 2011].

The following table shows the number of reduced samples and reduced features for each dataset. We used a binary ground truth without fractional values, so the number of reduced samples may be greater than $|\mathcal{P}|$.

Table 1: Size comparison between original and reduced datasets after compression (primal models). BLR stands for binary logistic regression.

DATASET	ORIG. SAMPLES	RED. SAMPLES	ORIG. FEATURES	RED. FEATURES
TITANIC	1309	1162	378	367
MNIST	70000	69952	784	715
ADULT (BLR)	48842	14723	105	102
ADULT (RIDGE)	48842	13329	104	102

Table 2: Size comparison between original and reduced datasets after compression (kernel models).

DATASET	DIMENSIONS OF K	DIMENSIONS OF $K'_{\text{LOGIT}}/K'_{\text{PRED}}$	DIMENSIONS OF K'_{QUAD}
ADULT (BLR)	4000×4000	2347×2170	2170×2170
ADULT (RIDGE)	4000×4000	2170×2170	2170×2170

To empirically verify that compression preserves the training objective and predictive behavior, we also compare the total objective value and maximum discrepancy in predicted probabilities between the original and lifted reduced solutions.

Table 3: Objective difference (absolute and relative) and train data prediction difference (absolute) between original and reduced datasets after compression. For ridge rows, the last column is the maximum prediction difference rather than the maximum predicted-probability difference.

DATASET	ABSOLUTE Δ OBJ	RELATIVE Δ OBJ	MAX ABSOLUTE $\Delta\hat{p}$
TITANIC	4.9659×10^{-3}	$1.3156 \times 10^{-3}\%$	4.7879×10^{-3}
MNIST	6.6412×10^{0}	$4.3050 \times 10^{-2}\%$	6.9964×10^{-2}
ADULT (PRIMAL BLR)	8.8778×10^{-2}	$5.3667 \times 10^{-4}\%$	6.7271×10^{-2}
ADULT (KERNEL BLR)	4.6141×10^{-1}	$3.3250 \times 10^{-2}\%$	1.2306×10^{-1}
ADULT (PRIMAL RIDGE)	9.3132×10^{-10}	$1.7662 \times 10^{-14}\%$	4.1105×10^{-12}
ADULT (KERNEL RIDGE)	1.1642×10^{-10}	$2.7003 \times 10^{-14}\%$	7.1167×10^{-8}

The residual discrepancies in Table 3 come from the logistic regression model's default convergence tolerance (tol $= 10^{-4}$) rather than from reduction. With the default lbfgs solver, convergence is limited by an internal function-evaluation cap. Rerunning with the newton-cg solver, which has no such cap, the maximum predicted-probability gap shrinks to 1.3×10^{-3} on MNIST as tolerance decreases to 10^{-6}, 2.1×10^{-8} on Adult (primal binary logistic) as tolerance decreases to 10^{-10}, and 2.6×10^{-5} on Adult (kernel binary logistic) as tolerance decreases to 10^{-12}.

Finally, we compare training runtime on CPU. Runtime is given in seconds and includes time taken for StandardScaler to fit and transform the data matrix (for Titanic and the primal binary logistic Adult setting).

Table 4: Effects of compression on training runtime (mean and sample standard deviation across 50 trials per dataset).

DATASET	TRAIN TIME (ORIG.)	COMPRESSION TIME	TRAIN TIME (RED.)
TITANIC	0.2995±0.1519	0.0187±0.0268	0.0465±0.0340
MNIST	279.3438±2.1568	14.1192±0.2971	257.1018±3.9239
ADULT (PRIMAL BLR)	0.6100±0.1315	0.3976±0.0415	0.2609±0.1136
ADULT (KERNEL BLR)	258.0607±22.3155	1.8078±0.0276	223.9466±21.6734
ADULT (PRIMAL RIDGE)	0.0683±0.0145	0.4416±0.0418	0.0275±0.0070
ADULT (KERNEL RIDGE)	1.6824±0.2581	0.9072±0.0567	0.3985±0.1249

Remark H.1. Primal ridge regression on Adult is the one setting where reduction does not pay off for a single fit. The reduction pays off only when it is reused across many fits, such as a regularization or

hyperparameter sweep. Additionally, primal binary logistic regression is roughly break-even for a single fit.

Remark H.2. Reduction affects two separate factors in the fit time: the cost of each solver iteration and the number of iterations the solver needs. On Titanic, reduction decreases both the iteration count (35 to 28) and the time per iteration (4.32×10^{-3}s to 2.61×10^{-3}s), so the gains compound. On MNIST, the reduction is modest, and both quantities remain similar (611 vs. 644 iterations; 0.472s vs. 0.431s per iteration). On Adult (kernel), reduction decreases the time per iteration (4.11×10^{-2}s to 1.53×10^{-2}s), consistent with the smaller reduced problem, but the solver requires more iterations (5674 to 12669), largely offsetting this gain. Thus, dimensional compression directly reduces the computational work per iteration, but does not necessarily reduce the number of iterations required by the numerical solver.

I Sensitivity to Noise and Sparsity

Color refinement merges rows and columns only under exact numerical equality, so we study two controlled regimes in which compression degrades:

- Continuous noise. After adding Gaussian noise with standard deviation 10^{-6} to every entry in the full Adult dataset predicting age, color refinement gives zero compression. Any floating-point perturbation, however small, defeats exact numerical equality.
- Sparsity. We generated a 20000×5000 binary matrix with each entry independently equal to 1 with probability equal to the stated density, together with uniformly random binary labels, and ran color refinement on it. At density 0.01 and higher, we observed no compression at all. At density 0.001, rows dropped to 19821. Below that, there is a sharp transition: density 0.0003 gives 12576 rows and 4969 columns, and density 0.0001 gives 3206 rows and 2250 columns.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describe the theoretical reduction framework and its applicability to differentiable convex ERM problems, and the empirical section supports the claimed practical effectiveness in structured datasets.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section (Appendix B) discussing key assumptions, including dependence on structural redundancy and convexity, as well as the scope of empirical evaluation and computational tradeoffs.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results include clearly stated assumptions, and complete proofs are provided in the appendix with appropriate cross-referencing.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper specifies datasets, preprocessing, model configurations, and evaluation procedures sufficient to reproduce the main experimental findings.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used in the experiments are publicly available. Full code and instructions necessary to reproduce the reported experiments are provided in the supplemental material.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides sufficient details on datasets, preprocessing, and training setup, with additional implementation details described in Section 4 and Appendix H.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports sample standard deviation for runtime experiments, states the source of variability (repeated trials), and explains how the error bars are computed.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the hardware and compute environment used for the experiments in Section 4.2.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work is primarily theoretical, with experiments conducted on publicly available benchmark datasets that do not contain sensitive or personally identifiable information. As such, it does not raise concerns related to privacy, human subjects, or societal impact under the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is foundational and does not target a specific application with direct societal impact.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks. Experiments use publicly available tabular datasets that do not contain sensitive or personally identifiable information.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and software used in the paper are publicly available, properly cited, and their licenses or terms of use are explicitly stated and properly respected.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets or released software artifacts.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.