Random Coordinate Descent Methods for Minimizing Decomposable Submodular Functions

Alina Ene A.ENE@DCS.WARWICK.AC.UK
Department of Computer Science and DIMAP, University of Warwick

Huy L. Nguyen HLNGUYEN@CS.PRINCETON.EDU
Simons Institute, University of California, Berkeley

Abstract

Submodular function minimization is a fundamental optimization problem that arises in several applications in machine learning and computer vision. The problem is known to be solvable in polynomial time, but general purpose algorithms have high running times and are unsuitable for large-scale problems. Recent work have used convex optimization techniques to obtain very practical algorithms for minimizing functions that are sums of "simple" functions. In this paper, we use random coordinate descent methods to obtain algorithms with faster *linear* convergence rates and cheaper iteration costs. Compared to alternating projection methods, our algorithms do not rely on full-dimensional vector operations and they converge in significantly fewer iterations.

1. Introduction

Over the past few decades, there has been a significant progress on minimizing submodular functions, leading to several polynomial time algorithms for the problem (Grötschel et al., 1981; Schrijver, 2000; Iwata, 2003; Fleischer & Iwata, 2003; Orlin, 2009). Despite this intense focus, the running times of these algorithms are high-order polynomials in the size of the data and designing faster algorithms remains a central and challenging direction in submodular optimization.

At the same time, technological advances have made it possible to capture and store data at an ever increasing rate and level of detail. A natural consequence of this "big data" phenomenon is that machine learning applications need to cope with data that is quite large and is growing at a fast pace. Thus there is an increasing need for algorithms that are fast and scalable.

The general purpose algorithms for submodular minimization are designed to provide worst-case guarantees even in settings where the only structure that one can exploit is submodularity. At the other extreme, graph cut algorithms are very efficient but they cannot handle more general submodular functions. In many applications, the functions strike a middle ground between these two extremes and it is becoming increasingly more important to use their special structure to obtain significantly faster algorithms.

Following (Kolmogorov, 2012; Stobbe & Krause, 2010; Jegelka et al., 2013; Nishihara et al., 2014a), we consider the problem of minimizing *decomposable* submodular functions that can be expressed as a sum of *simple* functions. We use the term simple to refer to functions F for which there is an efficient algorithm for minimizing $F+w$, where w is a linear function. We assume that we are given black-box access to these minimization procedures for simple functions.

Decomposable functions are a fairly rich class of functions and they arise in several applications in machine learning and computer vision. For example, they model higher-order potential functions for MAP inference in Markov random fields, the cost functions in SVM models for which the examples have only a small number of features, and the graph and hypergraph cut functions in image segmentation.

The recent work of (Jegelka et al., 2013; Kolmogorov, 2012; Stobbe & Krause, 2010) has developed several algorithms with very good empirical performance that exploit the special structure of decomposable functions. In particular, (Jegelka et al., 2013) have shown that the problem of minimizing decomposable submodular functions can be formulated as a *distance minimization* problem between two polytopes. This formulation, when coupled with powerful convex optimization techniques such as gradient de-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

scent or projection methods, yields algorithms that are very fast in practice and very simple to implement (Jegelka et al., 2013).

On the theoretical side, the convergence behaviour of these methods is not very well understood. Very recently, Nishihara *et al.* (2014a) have made a significant progress in this direction. Their work shows that the classical *alternating projections* method, when applied to the distance minimization formulation, converges at a *linear rate*.

Our contributions. In this work, we use random coordinate descent methods in order to obtain algorithms for minimizing decomposable submodular functions with faster convergence rates and cheaper iteration costs. We analyze a standard and an accelerated random coordinate descent algorithm and we show that they achieve linear convergence rates. Compared to alternating projection methods, our algorithms do not rely on full-dimensional vector operations and they are faster by a factor equal to the number of simple functions. Moreover, our accelerated algorithm converges in a much smaller number of iterations. We experimentally evaluate our algorithms on image segmentation tasks and we show that they perform very well and they converge much faster than the alternating projection method.

Submodular minimization. The first polynomial time algorithm for submodular optimization was obtained by Grötschel *et al.* (1981) using the ellipsoid method. There are several combinatorial algorithms for the problem (Schrijver, 2000; Iwata, 2003; Fleischer & Iwata, 2003; Orlin, 2009). Among the combinatorial methods, Orlin's algorithm (2009) achieves the best time complexity of $O(n^5T + n^6)$, where n is the size of the ground set and T is the maximum amount of time it takes to evaluate the function. Several algorithms have been proposed for minimizing decomposable submodular functions (Stobbe & Krause, 2010; Kolmogorov, 2012; Jegelka et al., 2013; Nishihara et al., 2014a). Stobbe and Krause (2010) use gradient descent methods with sublinear convergence rates for minimizing sums of concave functions applied to linear functions. Nishihara *et al.* (2014a) give an algorithm based on alternating projections that achieves a linear convergence rate.

1.1. Preliminaries and Background

Let V be a finite ground set of size n; without loss of generality, $V = \{1, 2, \ldots, n\}$. We view each point $w \in \mathbb{R}^n$ as a modular set function $w(A) = \sum_{i \in A} w_i$ on the ground set V.

A set function $F : 2^V \to \mathbb{R}$ is *submodular* if $F(A) + F(B) \geq F(A \cap B) + F(A \cup B)$ for any two sets $A, B \subseteq V$. A set function $F_i : 2^V \to \mathbb{R}$ is *simple* if there is a fast subroutine for minimizing $F_i + w$ for any modular function $w \in \mathbb{R}^n$.

In this paper, we consider the problem of minimizing a submodular function $F : 2^V \to \mathbb{R}$ of the form $F = \sum_{i=1}^{r} F_i$, where each function F_i is a simple submodular set function:

$$\min_{A \subseteq V} F(A) \equiv \min_{A \subseteq V} \sum_{i=1}^{r} F_i(A). \qquad \text{(DSM)}$$

We assume without loss of generality that the function F is normalized, i.e., $F(\emptyset) = 0$. Additionally, we assume we are given black-box access to oracles for minimizing $F_i + w$ for each function F_i in the decomposition and each $w \in \mathbb{R}^n$.

The *base polytope* $B(F)$ of F is defined as follows.

$$B(F) = \{w \in \mathbb{R}^n \mid w(A) \leq F(A) \text{ for all } A \subseteq V, \\ w(V) = F(V)\}.$$

The discrete problem (DSM)[1] admits an exact convex programming relaxation based on the Lovász extension of a submodular function. The Lovász extension f of F can be written as the support function of the base polytope $B(F)$:

$$f(x) = \max_{w \in B(F)} \langle w, x \rangle \quad \forall x \in \mathbb{R}^n.$$

Even though the base polytope $B(F)$ has exponentially many vertices, the Lovász extension f can be evaluated efficiently using the greedy algorithm of Edmonds (see for example (Schrijver, 2003)). Given any point $x \in \mathbb{R}^n$, Edmonds' algorithm evaluates $f(x)$ using $O(n \log n + nT)$ time, where T is the time needed to evaluate the submodular function F.

Lovász showed that a set function F is submodular if and only if its Lovász extension f is convex (Lovász, 1983). Thus we can relax the problem of minimizing F to the following non-smooth convex optimization problem:

$$\min_{x \in [0,1]^n} f(x) \equiv \min_{x \in [0,1]^n} \sum_{i=1}^{r} f_i(x),$$

where f_i is the Lovász extension of F_i.

The relaxation above is exact. Given a fractional solution x to the Lovász Relaxation, the best threshold set of x has cost at most $f(x)$.

An important drawback of the Lovász relaxation is that its objective function is not smooth. Following previous work (Jegelka et al., 2013; Nishihara et al., 2014a), we consider a proximal version of the problem ($\|\cdot\|$ denotes the ℓ_2-norm):

[1] DSM stands for decomposable submodular function minimization.

RCDM Algorithm for (Prox-DSM)

$\langle\langle$*We can take the initial point* y_0 *to be* $0\rangle\rangle$

Start with $y_0 = (y_0^{(1)}, \ldots, y_0^{(r)}) \in \mathcal{Y}$

In each iteration k ($k \geq 0$)

Pick an index $i_k \in \{1, 2, \ldots, r\}$ uniformly at random

$\langle\langle$*Update the block* $i_k\rangle\rangle$

$$y_{k+1}^{(i_k)} \leftarrow \arg\min_{y \in B(F_{i_k})} \Big(\left\langle \nabla_{i_k} g(y_k), y - y_k^{(i_k)} \right\rangle + \tfrac{L_{i_k}}{2} \left\| y - y_k^{(i_k)} \right\|^2 \Big)$$

Figure 1. Random block coordinate descent method for (Prox-DSM). It finds a solution to (Prox-DSM) given access to an oracle for $\min_{y \in B(F_i)} \left(\langle y, a \rangle + \|y\|^2 \right)$.

$$\min_{x \in \mathbb{R}^n} \left(f(x) + \frac{1}{2} \|x\|^2 \right) \equiv \min_{x \in \mathbb{R}^n} \sum_{i=1}^{r} \left(f_i(x) + \frac{1}{2r} \|x\|^2 \right). \tag{Proximal}$$

Given an optimal solution x to the proximal problem $\min_{x \in \mathbb{R}^n} \left(f(x) + \frac{1}{2} \|x\|^2 \right)$, we can construct an optimal solution to the discrete problem (DSM) by thresholding x at zero; more precisely, the set $\{v \colon x(v) \geq 0\}$ is an optimal solution to (DSM) (Proposition 8.6 in (Bach, 2011)).

Lemma 1 ((Jegelka et al., 2013))**.** *The dual of the proximal problem*

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{r} \left(f_i(x) + \frac{1}{2r} \|x\|^2 \right)$$

is the problem

$$\max_{y^{(1)} \in B(F_1), \ldots, y^{(r)} \in B(F_r)} -\frac{1}{2} \left\| \sum_{i=1}^{r} y^{(i)} \right\|^2.$$

The primal and dual variables are linked as $x = -\sum_{i=1}^{r} y^{(i)}$.

We write the dual proximal problem in the following equivalent form:

$$\min_{y^{(1)} \in B(F_1), \ldots, y^{(r)} \in B(F_r)} \left\| \sum_{i=1}^{r} y^{(i)} \right\|^2. \tag{Prox-DSM}$$

It follows from the discussion above that, given an optimal solution $y = (y^{(1)}, \ldots, y^{(r)})$ to (Prox-DSM), we can recover an optimal solution to (DSM) by thresholding $x = -\sum_{i=1}^{r} y^{(i)}$ at zero.

2. Random Coordinate Descent Algorithm

In this section, we give an algorithm for the problem (Prox-DSM) that is based on the random coordinate gradient descent method (RCDM) of (Nesterov, 2012). The algorithm is given in Figure 1. The algorithm is very easy to implement and it uses oracles for problems of the form $\min_{y \in B(F_i)} \left(\langle y, a \rangle + \|y\|^2 \right)$, where $i \in [r]$ and $a \in \mathbb{R}^n$. Since each function F_i is simple, we have such oracles that are very efficient.

In the remainder of this section, we analyze the convergence rate of the RCDM algorithm. We emphasize that the objective function of (Prox-DSM) is *not* strongly convex and thus we cannot use as a black-box Nesterov's analysis of the RCDM method for minimizing strongly convex functions. Instead, we exploit the special structure of the problem to achieve convergence guarantees that match the rate achievable for strong convex objectives with strong convexity parameter $1/(n^2 r)$. Our analysis shows that the RCDM algorithm is faster by a factor of r (same convergence rate but faster iterations) than the alternating projections algorithm from (Nishihara et al., 2014a).

Outline of the analysis: Our analysis has two main components. First, we build on the work of (Nishihara et al., 2014a) in order to prove a key theorem (Theorem 2). This theorem exploits the special structure of the (Prox-DSM) problem and it allows us to overcome the fact that the objective function of (Prox-DSM) is not strongly convex. Second, we modify Nesterov's analysis of the RCDM algorithm for minimizing strongly convex functions and we replace the strong convexity guarantee by the guarantee given by Theorem 2.

We start by introducing some notation; for the most part, we follow the notation of (Nesterov, 2012) and (Nishihara et al., 2014a). Let $\mathbb{R}^{nr} = \bigotimes_{i=1}^{r} \mathbb{R}^n$. We write a vector $y \in \mathbb{R}^{nr}$ as $y = (y^{(1)}, \ldots, y^{(r)})$, where each block $y^{(i)}$ is an n-dimensional vector. Let $\mathcal{Y} = \bigotimes_{i=1}^{r} B(F_i)$ be the constraint set of (Prox-DSM). Let $g : \mathbb{R}^{nr} \to \mathbb{R}$ be the objective function of (Prox-DSM): $g(y) = \left\| \sum_{i=1}^{r} y^{(i)} \right\|^2$. We use ∇g to denote the gradient of g, i.e., the (nr)-dimensional vector of partial derivatives. For each $i \in \{1, \ldots, r\}$, we use $\nabla_i g(y) \in \mathbb{R}^n$ to denote the i-th block of coordinates of $\nabla g(y)$.

Let $S \in \mathbb{R}^{n \times nr}$ be the following matrix:

$$S = \frac{1}{\sqrt{r}} \Big[\underbrace{I_n I_n \cdots I_n}_{r \text{ times}} \Big].$$

Note that $g(y) = r \|Sy\|^2$ and $\nabla g(y) = 2rS^T Sy$. Additionally, for each $i \in \{1, 2, \ldots, r\}$, $\nabla_i g$ is Lipschitz continuous with constant $L_i = 2$:

$$\|\nabla_i g(x) - \nabla_i g(y)\| \leq L_i \left\| x^{(i)} - y^{(i)} \right\|, \tag{1}$$

for all vectors $x, y \in \mathbb{R}^{nr}$ that differ only in block i.

Our first step is to prove the following key theorem that builds on the work of (Nishihara et al., 2014a).

Theorem 2. *Let $y \in \mathcal{Y}$ be a feasible solution to (Prox-DSM). Let y^* be an optimal solution to (Prox-DSM) that minimizes $\|y - y^*\|$. We have*

$$\|S(y - y^*)\| \geq \frac{1}{nr} \|y - y^*\|.$$

The proof of Theorem 2 uses the following key result from (Nishihara et al., 2014b). We will need the following definitions from (Nishihara et al., 2014b).

Let $d(K_1, K_2) = \inf\{\|k_1 - k_2\| : k_1 \in K_1, k_2 \in K_2\}$ be the distance between sets K_1 and K_2. Let $\mathcal{P}$ and $\mathcal{Q}$ be two closed convex sets in $\mathbb{R}^d$. Let $E \subseteq \mathcal{P}$ and $H \subseteq \mathcal{Q}$ be the sets of closest points

$$E = \{p \in \mathcal{P} \colon d(p, \mathcal{Q}) = d(\mathcal{P}, \mathcal{Q})\},$$
$$H = \{q \in \mathcal{Q} \colon d(q, \mathcal{P}) = d(\mathcal{P}, \mathcal{Q})\}.$$

Since $\mathcal{P}$ and $\mathcal{Q}$ are convex, for each point in $p \in E$, there is a unique point $q \in H$ such that $d(p, q) = d(\mathcal{P}, \mathcal{Q})$ and vice versa. Let $v = \Pi_{\mathcal{Q}-\mathcal{P}} 0$, where $\Pi_{\mathcal{Q}-\mathcal{P}}$ is the projection operator onto $\mathcal{Q}-\mathcal{P}$; note that $H = E+v$. Let $\mathcal{Q}' = \mathcal{Q}-v$; $\mathcal{Q}'$ is a translated version of $\mathcal{Q}$ and it intersects $\mathcal{P}$ at E. Let

$$\kappa_* = \sup_{x \in (\mathcal{P} \cup \mathcal{Q}') \setminus E} \frac{d(x, E)}{\max\{d(x, \mathcal{P}), d(x, \mathcal{Q}')\}}.$$

By combining Corollary 5 and Proposition 11 from (Nishihara et al., 2014b), we obtain the following theorem.

Theorem 3 ((Nishihara et al., 2014a))**.** *If $\mathcal{P}$ is the polyhedron $\bigotimes_{i=1}^r B(F_i)$ and $\mathcal{Q}$ is the polyhedron $\left\{y \in \mathbb{R}^{nr} \colon \sum_{i=1}^r y^{(i)} = 0\right\}$, we have $\kappa_* \leq nr$.*

Now we are ready to prove Theorem 2. Let

$$\mathcal{P} = \bigotimes_{i=1}^r B(F_i) = \mathcal{Y},$$
$$\mathcal{Q} = \left\{y \in \mathbb{R}^{nr} \colon \sum_{i=1}^r y^{(i)} = 0\right\} = \{y \in \mathbb{R}^{nr} \colon Sy = 0\}.$$

We define $\mathcal{Q}'$ and κ_* as above.

Let y and y^* be the two points in the statement of the theorem. Note that $y \in \mathcal{P}$ and $y^* \in E$, since E is the set of all optimal solutions to (Prox-DSM). We may assume that $y \notin E$, since otherwise the theorem trivially holds. Since $y \in \mathcal{P} \setminus E$, we have

$$\kappa_* \geq \frac{d(y, E)}{d(y, \mathcal{Q}')}.$$

Since y^* is an optimal solution that is closest to y, we have $d(y, E) = \|y - y^*\|$. Using the fact that the rows of S form a basis for the orthogonal complement of $\mathcal{Q}$, we can show that $d(y, \mathcal{Q}') = \|S(y - y^*)\|$.

Therefore

$$\kappa_* \geq \frac{\|y - y^*\|}{\|S(y - y^*)\|}.$$

Theorem 2 now follows from Theorem 3.

In the remainder of this section, we use Nesterov's analysis (Nesterov, 2012) in conjunction with Theorem 2 in order to show that the RCDM algorithm converges at a linear rate. Recall that E is the set of all optimal solutions to (Prox-DSM).

Theorem 4. *After $(k+1)$ iterations of the RCDM algorithm, we have*

$$\mathbb{E}\left[d(y_k, E)^2 + g(y_{k+1}) - g(y^*)\right] \leq \left(1 - \frac{2}{n^2 r^2 + r}\right)^{k+1} \left(d(y_0, E)^2 + g(y_0) - g(y^*)\right),$$

where $y^* \in E$ is an arbitrary optimal solution to (Prox-DSM).

We devote the rest of this section to the proof of Theorem 4. We recall the following well-known lemma, which we refer to as the first-order optimality condition.

Lemma 5 (Theorem 2.2.5 in (Nesterov, 2004))**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable convex function and let $Q \subseteq \mathbb{R}^d$ be a closed convex set. A point $x^* \in \mathbb{R}^d$ is a solution to the problem $\min_{x \in Q} f(x)$ if and only if*

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0$$

for all $x \in Q$.

It follows from the first-order optimality condition for $y_{k+1}^{(i_k)}$ that, for any $z \in B(F_{i_k})$,

$$\left\langle \nabla_{i_k} g(y_k) + L_{i_k}\left(y_{k+1}^{(i_k)} - y_k^{(i_k)}\right), z - y_{k+1}^{(i_k)} \right\rangle \geq 0. \quad (2)$$

We show in the supplement that

$$g(y_{k+1}) = g(y_k) + \left\langle \nabla_{i_k} g(y_k), y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle + \frac{L_{i_k}}{2} \left\| y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\|^2. \quad (3)$$

Let $y^* = \arg\min_{y \in E} \|y - y_k\|$ be the optimal solution that is closest to y_k. Using (2) and (3), we show in the supplement that

$$\|y_{k+1} - y^*\|^2 \leq \|y_k - y^*\|^2 + \frac{2}{L_{i_k}} \left\langle \nabla_{i_k} g(y_k), (y^*)^{(i_k)} - y_k^{(i_k)} \right\rangle - \frac{2}{L_{i_k}} \left(g(y_{k+1}) - g(y_k)\right). \quad (4)$$

If we rearrange the terms of the inequality (4), take expectation over i_k, and substitute $L_{i_k} = 2$, we obtain

$$\mathbb{E}_{i_k}\left[\|y_{k+1} - y^*\|^2 + g(y_{k+1}) - g(y^*)\right]$$

$$\leq \|y_k - y^*\|^2 + g(y_k) - g(y^*) + \frac{1}{r}\langle \nabla g(y_k), y^* - y_k\rangle. \tag{5}$$

We can upper bound $\langle \nabla g(y_k), y^* - y_k\rangle$ as follows.

$$\begin{aligned}
&\langle \nabla g(y_k), y^* - y_k\rangle = 2r\left\langle S^T S y_k, y^* - y_k\right\rangle \\
&= r\left\langle S^T S y_k + S^T S y^*, y^* - y_k\right\rangle + \\
&\quad r\left\langle S^T S y_k - S^T S y^*, y^* - y_k\right\rangle \\
&= r\left\langle S^T S y_k + S^T S y^*, y^* - y_k\right\rangle - r\|S(y_k - y^*)\|^2 \\
&= r\langle S(y_k + y^*), S(y^* - y_k)\rangle - r\|S(y_k - y^*)\|^2 \\
&= (g(y^*) - g(y_k)) - r\|S(y_k - y^*)\|^2 \\
&\leq (g(y^*) - g(y_k)) - \frac{1}{n^2 r}\|y_k - y^*\|^2. \quad \text{(By Theorem 2)}
\end{aligned} \tag{6}$$

On the first and fifth lines, we have used the fact that $\nabla g(z) = 2rS^T Sz$ and $g(z) = r\|Sz\|^2$ for any $z \in \mathbb{R}^{nr}$. On the last line, we have used Theorem 2.

Since y^* is an optimal solution to (Prox-DSM), the first-order optimality condition gives us that

$$\langle \nabla g(y^*), y^* - y_k\rangle = 2r\langle S^T S y^*, y^* - y_k\rangle \leq 0. \tag{7}$$

Using the inequality above, we can also upper bound $\langle \nabla g(y_k), y^* - y_k\rangle$ as follows.

$$\begin{aligned}
&\langle \nabla g(y_k), y^* - y_k\rangle = 2r\langle S^T S y_k, y^* - y_k\rangle \\
&= 2r\langle S^T S y^*, y^* - y_k\rangle + 2r\langle S^T S y_k - S^T S y^*, y^* - y_k\rangle \\
&= 2r\langle S^T S y^*, y^* - y_k\rangle - 2r\|S(y_k - y^*)\|^2 \\
&\overset{(7)}{\leq} -2r\|S(y_k - y^*)\|^2 \\
&\leq -\frac{2}{n^2 r}\|y_k - y^*\|^2. \quad \text{(By Theorem 2)}
\end{aligned} \tag{8}$$

By taking $\frac{2}{n^2 r+1} \times (6) + \left(1 - \frac{2}{n^2 r+1}\right) \times (8)$, we obtain

$$\begin{aligned}
&\langle \nabla g(y_k), y^* - y_k\rangle \leq \\
&-\frac{2}{n^2 r + 1}\left(g(y_k) - g(y^*) + \|y_k - y^*\|^2\right).
\end{aligned} \tag{9}$$

By (5) and (9),

$$\begin{aligned}
&\mathbb{E}_{i_k}\left[\|y_{k+1} - y^*\|^2 + g(y_{k+1}) - g(y^*)\right] \\
&\leq \left(1 - \frac{2}{n^2 r^2 + r}\right)\left(g(y_k) - g(y^*) + \|y_k - y^*\|^2\right).
\end{aligned}$$

Note that $d(y_{k+1}, E)^2 \leq \|y_{k+1} - y^*\|^2$ and $d(y_k, E)^2 = \|y_k - y^*\|^2$. Therefore

$$\begin{aligned}
&\mathbb{E}_{i_k}\left[d(y_{k+1}, E)^2 + g(y_{k+1}) - g(y^*)\right] \\
&\leq \left(1 - \frac{2}{n^2 r^2 + r}\right)\left(d(y_k, E)^2 + g(y_k) - g(y^*)\right).
\end{aligned}$$

APPROX algorithm applied to (Prox-DSM)

Start with $z_0 = (z_0^{(1)}, \ldots, z_0^{(r)}) \in \mathcal{Y}$, $\theta_0 = \frac{1}{r}$, $u_0 = 0$
In each iteration k ($k \geq 0$)
  Generate a random set of blocks R_k where each block is included independently with probability $\frac{1}{r}$
  $u_{k+1} \leftarrow u_k$, $z_{k+1} \leftarrow z_k$
  For each $i \in R_k$
    $t_k^{(i)} \leftarrow \underset{t + z_k^{(i)} \in B(F_{i_k})}{\arg\min}\left(\left\langle \nabla_i g\left(\theta_k^2 u_k + z_k\right), t\right\rangle + 2r\theta_k \|t\|^2\right)$
    $z_{k+1}^{(i)} \leftarrow z_k^{(i)} + t_k^{(i)}$
    $u_{k+1}^{(i)} \leftarrow u_k^{(i)} - \frac{1 - r\theta_k}{\theta_k^2} t_k^{(i)}$
  $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
Return $\theta_k^2 u_{k+1} + z_{k+1}$

Figure 2. The APPROX algorithm of (Fercoq & Richtárik, 2013) applied to (Prox-DSM). It finds a solution to (Prox-DSM) given access to an oracle for $\min_{y \in B(F_i)}\left(\langle y, a\rangle + \|y\|^2\right)$.

ACDM Algorithm for (Prox-DSM)

⟪We can take the initial point y_0 to be 0⟫
Start with $y_0 = (y_0^{(1)}, \ldots, y_0^{(r)}) \in \mathcal{Y}$
In each epoch ℓ ($\ell \geq 0$)
  Run the algorithm in Figure 2 for $(4nr^{3/2} + 1)$ iterations with y_ℓ as its starting point ($z_0 = y_\ell$)
  Let $y_{\ell+1}$ be the vector returned by the algorithm

Figure 3. Accelerated block coordinate descent method for (Prox-DSM). It finds a solution to (Prox-DSM) given access to an oracle for $\min_{y \in B(F_i)}\left(\langle y, a\rangle + \|y\|^2\right)$.

By taking expectation over $\xi = (i_1, \ldots, i_k)$, we get

$$\begin{aligned}
&\mathbb{E}_\xi\left[d(y_{k+1}, E)^2 + g(y_{k+1}) - g(y^*)\right] \\
&\leq \left(1 - \frac{2}{n^2 r^2 + r}\right)^{k+1}\left(d(y_0, E)^2 + g(y_0) - g(y^*)\right).
\end{aligned}$$

Therefore the proof of Theorem 4 is complete.

3. Accelerated Coordinate Descent Algorithm

In this section, we give an accelerated random coordinate descent (ACDM) algorithm for (Prox-DSM). The algorithm uses the APPROX algorithm of Fercoq and Richtárik (2013) as a subroutine. The APPROX algorithm (Algorithm 2 in (Fercoq & Richtárik, 2013)), when applied to the (Prox-DSM) problem, yields the algorithm in Figure 2. The ACDM algorithm runs in a sequence of epochs (see Figure 3). In each epoch, the algorithm starts with the solution of the previous epoch and it runs the APPROX algorithm for $\Theta(nr^{3/2})$ iterations. The solution constructed by the APPROX algorithm will be the starting point of the next epoch. Note that, for each i, the gradient $\nabla_i g(y) =$

$2\sum_j y^{(j)}$ can be easily maintained at a cost of $O(n)$ per block update, and thus the iteration cost is dominated by the time to compute the projection.

In the remainder of this section, we use the analysis of (Fercoq & Richtárik, 2013) together with Theorem 2 in order to show that the ACDM algorithm converges at a linear rate. We follow the notation used in Section 2.

Theorem 6. *After ℓ epochs of the ACDM algorithm (equivalently, $(4nr^{3/2}+1)\ell$ iterations), we have*

$$\mathbb{E}[g(y_{\ell+1}) - g(y^*)] \leq \frac{1}{2^{\ell+1}}(g(y_0) - g(y^*)).$$

In the following lemma, we show that the objective function of (Prox-DSM) satisfies Assumption 1 in (Fercoq & Richtárik, 2013) and thus the convergence analysis given in (Fercoq & Richtárik, 2013) can be applied to our setting.

Lemma 7. *Let $R \subseteq \{1,2,\ldots,r\}$ be a random subset of coordinate blocks with the property that each $i \in \{1,2,\ldots,r\}$ is in R independently at random with probability $1/r$. Let x and h be two vectors in $\mathbb{R}^{nr}$. Let h_R be the vector in $\mathbb{R}^{nr}$ such that $(h_R)^{(i)} = h^{(i)}$ for each block $i \in R$ and $(h_R)^{(i)} = 0$ otherwise. We have*

$$\mathbb{E}\left[g\left(x + h_R\right)\right] \leq g(x) + \frac{1}{r}\left\langle \nabla g(x), h\right\rangle + \frac{2}{r}\left\|h\right\|^2.$$

Proof: We have

$$\begin{aligned}
&\mathbb{E}\left[g\left(x + h_R\right)\right] = \mathbb{E}\left[r\left\|S(x+h_R)\right\|^2\right] \\
&= \mathbb{E}\left[r\left\|Sx\right\|^2 + r\left\|Sh_R\right\|^2 + 2r\left\langle Sx, Sh_R\right\rangle\right] \\
&= \mathbb{E}\left[r\left\|Sx\right\|^2 + r\left\|Sh_R\right\|^2 + 2r\left\langle S^TSx, h_R\right\rangle\right] \\
&= \mathbb{E}\left[g(x) + \left\|\sum_{i=1}^r h_R^{(i)}\right\|^2 + \left\langle \nabla g(x), h_R\right\rangle\right] \\
&= g(x) + \frac{1}{r^2}\sum_{i\neq j}\langle h^{(i)}, h^{(j)}\rangle + \frac{1}{r}\sum_{i=1}^r\left\|h^{(i)}\right\|^2 + \\
&\quad \frac{1}{r}\langle \nabla g(x), h\rangle \\
&\leq g(x) + \frac{2}{r}\sum_{i=1}^r\left\|h^{(i)}\right\|^2 + \frac{1}{r}\langle \nabla g(x), h\rangle.
\end{aligned}$$

□

Lemma 7 together with Theorem 3 in (Fercoq & Richtárik, 2013) give us the following theorem.

Theorem 8 (Theorem 3 of (Fercoq & Richtárik, 2013))**.** *Consider iteration k of the APPROX algorithm (see Figure 2). Let $y_k = \theta_k^2 u_{k+1} + z_{k+1}$. Let $y^* = \arg\min_{y\in E}\|y - z_0\|$ is the optimal solution that is closest to z_0. We have*

$$\mathbb{E}[g(y_k) - g(y^*)] \leq \frac{4r^2}{(k-1+2r)^2}\cdot \left(\left(1-\frac{1}{r}\right)(g(z_0)-g(y^*)) + 2\left\|z_0 - y^*\right\|^2\right).$$

Proof: It follows from Lemma 7 that the objective function g of (Prox-DSM) and the random blocks R_k used by the APPROX algorithm satisfy Assumption 1 in (Fercoq & Richtárik, 2013) with $\tau = 1$ and $\nu_i = 4$ for each $i \in \{1,2,\ldots,r\}$. Thus we can apply Theorem 3 in (Fercoq & Richtárik, 2013). □

Consider an epoch ℓ. Let $y_{\ell+1}$ be the solution constructed by the APPROX algorithm after $4nr^{3/2}+1$ iterations, starting with y_ℓ. Let $y^* = \arg\min_{y\in E}\|y - y_\ell\|$ be the optimal solution that is closest to y_ℓ. Let ξ_ℓ denote the random choices made during epoch ℓ. By Theorem 8,

$$\begin{aligned}
&\mathbb{E}_{\xi_\ell}[g(y_{\ell+1}) - g(y^*)] \leq \frac{4r^2}{(4nr^{3/2}+2r)^2}\cdot \\
&\left(\left(1-\frac{1}{r}\right)(g(y_\ell)-g(y^*)) + 2\left\|y_\ell - y^*\right\|^2\right) \\
&\leq \frac{1}{(2nr^{1/2}+1)^2}\left(g(y_\ell) - g(y^*) + 2\left\|y_\ell - y^*\right\|^2\right).
\end{aligned}$$

We also have

$$\begin{aligned}
&g(y_\ell) = g(y^*) + \langle \nabla g(y^*), y_\ell - y^*\rangle + \\
&\int_0^1 \langle \nabla g(y^* + t(y_\ell - y^*)) - \nabla g(y^*), y_\ell - y^*\rangle dt \\
&\geq g(y^*) + \int_0^1 \langle \nabla g(y^* + t(y_\ell - y^*)) - \nabla g(y^*), y_\ell - y^*\rangle dt \\
&= g(y^*) + \int_0^1 2tr\left\|S(y_\ell - y^*)\right\|^2 dt \\
&= g(y^*) + r\left\|S(y_\ell - y^*)\right\|^2 \\
&\geq g(y^*) + \frac{1}{n^2 r}\left\|y_\ell - y^*\right\|^2. \qquad \text{(By Theorem 2)}
\end{aligned}$$

In the second line, we have used the first-order optimality condition for y^* (Lemma 5). In the last line, we have used Theorem 2.

Therefore

$$\left\|y_\ell - y^*\right\|^2 \leq n^2 r(g(y_\ell) - g(y^*)),$$

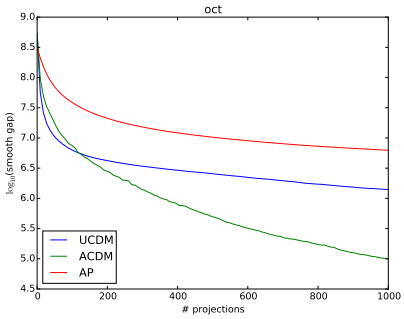
and hence

$$\begin{aligned}
\mathbb{E}_{\xi_\ell}[g(y_{\ell+1}) - g(y^*)] &\leq \frac{2n^2r+1}{(2nr^{1/2}+1)^2}(g(y_\ell) - g(y^*)) \\
&\leq \frac{1}{2}(g(y_\ell) - g(y^*)).
\end{aligned}$$

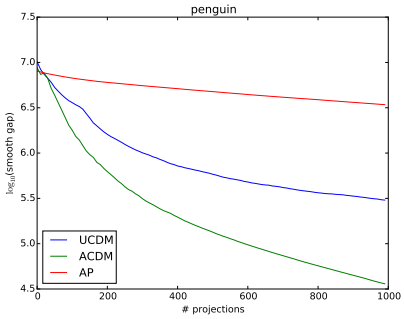
Let $\xi = (\xi_0, \ldots, \xi_\ell)$ be the random choices made during the epochs 0 to ℓ. We have

$$\mathbb{E}_{\xi}[g(y_{\ell+1}) - g(y^*)] \leq \frac{1}{2^{\ell+1}}(g(y_0) - g(y^*)).$$

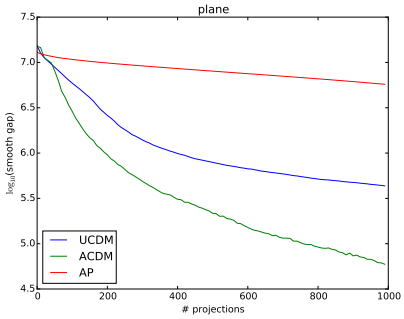
This completes the proof of Theorem 6 and the convergence analysis for the ACDM algorithm.



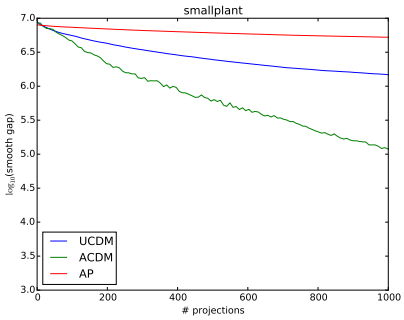
(a) Smooth gaps - Octopus



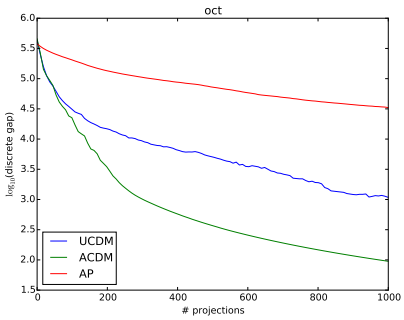
(b) Smooth gaps - Penguin



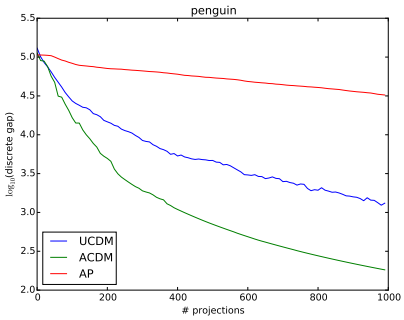
(c) Smooth gaps - Plane



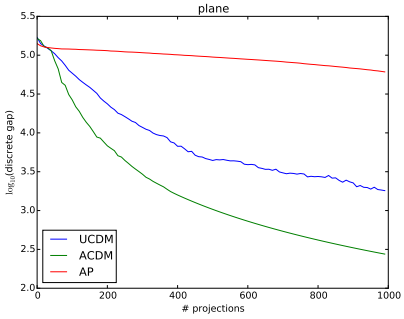
(d) Smooth gaps - Small plant



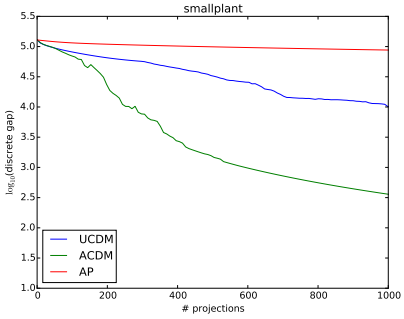
(e) Discrete gaps - Octopus



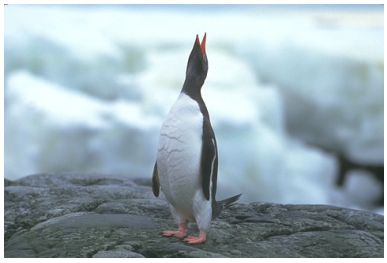
(f) Discrete gaps - Penguin



(g) Discrete gaps - Plane



(h) Discrete gaps - Small plant



(i) Penguin

Figure 4. Comparison of the convergence of the three algorithms (UCDM, ACDM, AP) on four image segmentation instances.

(a) ACDM 1
$\nu_s = 8.74 \cdot 10^6$
$\nu_d = 1.12 \cdot 10^5$

(b) ACDM 20
$\nu_s = 7.65 \cdot 10^6$
$\nu_d = 8.86 \cdot 10^4$

(c) ACDM 100
$\nu_s = 1.78 \cdot 10^6$
$\nu_d = 1.65 \cdot 10^4$

(d) AP 1
$\nu_s = 8.04 \cdot 10^6$
$\nu_d = 1.06 \cdot 10^5$

(e) AP 20
$\nu_s = 7.65 \cdot 10^6$
$\nu_d = 1.05 \cdot 10^5$

(f) AP 100
$\nu_s = 6.72 \cdot 10^6$
$\nu_d = 8.31 \cdot 10^4$

Figure 5. Penguin segmentation results for the fastest (ACDM) and slowest (AP) algorithms, after 1, 20, and 100 projections. The ν_s and ν_d values are the smooth and discrete duality gaps.

4. Experiments

Algorithms. We empirically evaluate and compare the following algorithms: the RCDM described in Section 2, the ACDM described in Section 3, and the alternating projections (AP) algorithm of (Nishihara et al., 2014a). The AP algorithm solves the following best approximation problem that is equivalent to (Prox-DSM):

$$\min_{a\in\mathcal{A},y\in\mathcal{Y}} \|a-y\|^2 \qquad \text{(Best-Approx)}$$

where $\mathcal{A} = \left\{(a^{(1)}, a^{(2)}, \ldots, a^{(r)}) \in \mathbb{R}^{nr} \colon \sum_{i=1}^r a^{(i)} = 0\right\}$ and $\mathcal{Y} = \bigotimes_{i=1}^r B(F_i)$.

The AP algorithm starts with a point $a_0 \in \mathcal{A}$ and it iteratively constructs a sequence $\{(a_k, y_k)\}_{k\geq 0}$ by projecting onto $\mathcal{A}$ and $\mathcal{Y}$: $y_k = \Pi_{\mathcal{Y}}(a_k)$, $a_{k+1} = \Pi_{\mathcal{A}}(y_k)$.

$\Pi_K(\cdot)$ is the projection operator onto K, that is, $\Pi_K(x) = \arg\min_{z\in K} \|x-z\|$. Since $\mathcal{A}$ is a subspace, it is straightforward to project onto $\mathcal{A}$. The projection onto $\mathcal{Y}$ can be implemented using the oracles for the projections $\Pi_{B(F_i)}$ onto the base polytopes of the functions F_i.

For all three algorithms, the iteration cost is dominated by the cost of projecting onto the base polytopes $B(F_i)$. Therefore the total number of such projections is a suitable measure for comparing the algorithms. In each iteration, the RCDM algorithm performs a single projection for a random block i and the ACDM algorithm performs a single projection in expectation. The AP algorithm performs r projections in each iteration, one for each block.

Image Segmentation Experiments. We evaluate the algorithms on graph cut problems that arise in image segmentation or MAP inference tasks in Markov Random Fields. Our experimental setup is similar to that of (Jegelka et al., 2013). We set up the image segmentation problems on a 8-neighbor grid graph with unary potentials derived from Gaussian Mixture Models of color features (Rother et al., 2004). The weight of a graph edge (i,j) between pixels i and j is a function of $\exp(-\|v_i - v_j\|^2)$, where v_i is the RGB color vector of pixel i. The optimization problem that we solve for each segmentation task is a cut problem on the grid graph.

Remarks on the ACDM algorithm: We emphasize that, in our experiments, the number of iterations is smaller than the size of an epoch of the ACDM algorithm, and thus there are no restarts. We have run experiments where we restarted the ACDM algorithm after a much smaller number of iterations, and we have found that this approach leads to a slower convergence rate.

Our implementation of the ACDM algorithm uses random permutations of the blocks instead of picking a block independently and uniformly at random in each iteration. Since the ACDM algorithm is randomized, we have run the algorithm several times. We have found that the difference in the duality gaps between different runs is very small and we have chosen to report the results of a single run instead of the averages.

Function decomposition: We partition the edges of the grid into a small number of *matchings* and we decompose the function using the cut functions of these matchings. Note that it is straightforward to project onto the base polytopes of such functions using a sequence of projections onto line segments.

Duality gaps: We evaluate the convergence behaviours of the algorithms using the following measures. Let y be a feasible solution to the dual of the proximal problem (Proximal). The solution $x = -\sum_{i=1}^r y^{(i)}$ is a feasible solution for the proximal problem. We define the *smooth duality gap* to be the difference between the objective values of the primal solution x and the dual solution y: $\nu_s = \left(f(x) + \frac{1}{2}\|x\|^2\right) - \left(-\frac{r}{2}\|Sy\|^2\right)$. We use the pool adjacent violators algorithm to search for an improvement to the smooth duality gap; we use the same approach as the one described in (Jegelka et al., 2013). Additionally, we compute a discrete duality gap for the discrete problem (DSM) and the dual of its Lovász relaxation; the latter is the problem $\max_{z\in B(F)}(z)_-(V)$, where $(z)_- = \min\{z,0\}$ applied elementwise (Jegelka et al., 2013). The best level set S_x of the proximal solution $x = -\sum_{i=1}^r y^{(i)}$ is a solution to the discrete problem (DSM). The solution $z = -x = \sum_{i=1}^r y^{(i)}$ is a feasible solution for the dual of the Lovász relaxation. We define the *discrete duality gap* to be the difference between the objective values of these solutions: $\nu_d(x) = F(S_x) - (-x)_-(V)$.

We evaluated the algorithms on four image segmentation instances[2] (Jegelka & Bilmes, 2011; Rother et al., 2004). Figure 4 shows the smooth and discrete duality gaps on the four instances. Figure 5 shows some segmentation results for one of the instances.

Acknowledgements. We thank Stefanie Jegelka for providing us with some of the data used in our experiments.

[2]The data is available at `http://melodi.ee.washington.edu/~jegelka/cc/index.html` and `http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm`

References

Bach, Francis. Learning with submodular functions: A convex optimization perspective. *ArXiv preprint arXiv:1111.6453*, 2011.

Fercoq, Olivier and Richtárik, Peter. Accelerated, parallel and proximal coordinate descent. *ArXiv preprint arXiv:1312.5799*, 2013.

Fleischer, Lisa and Iwata, Satoru. A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discrete Applied Mathematics*, 131(2):311–322, 2003.

Grötschel, Martin, Lovász, László, and Schrijver, Alexander. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

Iwata, Satoru. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32 (4):833–840, 2003.

Jegelka, Stefanie and Bilmes, Jeff. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1897–1904. IEEE, 2011.

Jegelka, Stefanie, Bach, Francis, and Sra, Suvrit. Reflection methods for user-friendly submodular optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1313–1321, 2013.

Kolmogorov, Vladimir. Minimizing a sum of submodular functions. *Discrete Applied Mathematics*, 160(15): 2246–2258, 2012.

Lovász, László. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pp. 235–257. Springer, 1983.

Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

Nesterov, Yurii. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Nishihara, Robert, Jegelka, Stefanie, and Jordan, Michael I. On the convergence rate of decomposable submodular function minimization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 640–648, 2014a.

Nishihara, Robert, Jegelka, Stefanie, and Jordan, Michael I. On the convergence rate of decomposable submodular function minimization. *ArXiv preprint arXiv:1406.6474*, 2014b.

Orlin, James B. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

Rockafellar, R Tyrrell. *Convex analysis*. Number 28 in Princeton Mathematical Series. Princeton university press, 1970.

Rother, Carsten, Kolmogorov, Vladimir, and Blake, Andrew. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

Schrijver, Alexander. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

Schrijver, Alexander. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

Stobbe, Peter and Krause, Andreas. Efficient minimization of decomposable submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2208–2216, 2010.

A. Proof of Lemma 1

By the definition of the Lovász extension, for each $i \in [r]$, we have

$$f_i(x) = \max_{y^{(i)} \in B(F_i)} \langle y^{(i)}, x \rangle.$$

Therefore

$$\begin{aligned}
&\min_{x \in \mathbb{R}^n} \sum_{i=1}^{r} \left(f_i(x) + \frac{1}{2r} \|x\|^2 \right) \\
&= \min_{x \in \mathbb{R}^n} \sum_{i=1}^{r} \left(\max_{y^{(i)} \in B(F_i)} \langle y^{(i)}, x \rangle + \frac{1}{2r} \|x\|^2 \right) \\
&= \min_{x \in \mathbb{R}^n} \max_{y^{(1)} \in B(F_1), \ldots, y^{(r)} \in B(F_r)} \sum_{i=1}^{r} \left(\langle y^{(i)}, x \rangle + \frac{1}{2r} \|x\|^2 \right) \\
&= \max_{y^{(1)} \in B(F_1), \ldots, y^{(r)} \in B(F_r)} \min_{x \in \mathbb{R}^n} \sum_{i=1}^{r} \left(\langle y^{(i)}, x \rangle + \frac{1}{2r} \|x\|^2 \right) \\
&= \max_{y^{(1)} \in B(F_1), \ldots, y^{(r)} \in B(F_r)} -\frac{1}{2} \left\| \sum_{i=1}^{r} y^{(i)} \right\|^2 .
\end{aligned}$$

On the third line, we have used the fact that the function $\langle y, x \rangle + (1/2r) \|x\|^2$ is convex in x and linear in y, which allows us to exchange the min and the max (see for example Corollary 37.3.2 in Rockafellar (Rockafellar, 1970)). On the fourth line, we have used the fact that the minimum is achieved at $x = -\sum_{i=1}^{r} y^{(i)}$.

B. Proofs omitted from Section 2

If $x \in \mathbb{R}^{nr}$ and $\mathcal{X}$ is a subspace of $\mathbb{R}^{nr}$, we let $\Pi_{\mathcal{X}}(x)$ denote the projection of x on $\mathcal{X}$, that is, $\Pi_{\mathcal{X}}(x) = \arg\min_{z \in \mathbb{R}^{nr}} \|x - z\|$. We let $\mathcal{X}^{\perp}$ denote the orthogonal complement of the subspace $\mathcal{X}$.

Proposition 9. *For any point $x \in \mathbb{R}^{nr}$, $\Pi_{\mathcal{Q}^{\perp}}(x) = S^T S x$ and thus $\Pi_{\mathcal{Q}}(x) = x - S^T S x$.*

Proof: Since $\mathcal{Q}$ is the null space of S, $\mathcal{Q}^{\perp}$ is the row space of S. Since the rows of S are orthonormal, they form a basis for $\mathcal{Q}^{\perp}$. Therefore, if we let $v_1, \ldots, v_n$ denote the rows of S, we have

$$\Pi_{\mathcal{Q}^{\perp}}(x) = \sum_{i=1}^{n} \langle x, v_i \rangle v_i = S^T S x.$$

□

Proposition 10. *The set of all optimal solutions to (Prox-DSM) is equal to E.*

Proof: We have

$$\begin{aligned}
d(\mathcal{P}, \mathcal{Q}) &= \min_{y \in \mathcal{P}} \|y - \Pi_{\mathcal{Q}}(y)\| \\
&= \min_{y \in \mathcal{P}} \left\| S^T S y \right\| \qquad \langle\langle \textit{By Proposition 9} \rangle\rangle \\
&= \min_{y \in \mathcal{P}} \|S y\| .
\end{aligned}$$

Since (Prox-DSM) is the problem $\min_{y \in \mathcal{P}} r \|Sy\|^2$, E is the set of all optimal solutions to (Prox-DSM). □

Proposition 11. *Let $y \in \mathbb{R}^{nr}$ and let $p \in E$. We have $d(y, \mathcal{Q}') = \|S(y - p)\|$.*

Proof: Since $\mathcal{Q}' = \mathcal{Q} - v$, we have

$$\begin{aligned}
d(y, \mathcal{Q}') &= d(y + v, \mathcal{Q}) \\
&= \|\Pi_{\mathcal{Q}^{\perp}}(y + v)\| \\
&= \left\| S^T S (y + v) \right\| \qquad \langle\langle \textit{By Proposition 9} \rangle\rangle \\
&= \left\| S^T S (y - S^T S p) \right\| \qquad \langle\langle \textit{Since } v = -S^T S p \rangle\rangle \\
&= \left\| S^T S (y - p) \right\| \qquad \langle\langle \textit{Since } S S^T = I_n \rangle\rangle \\
&= \|S(y - p)\| .
\end{aligned}$$

□

Proof of Equation (3): We have

$$\begin{aligned}
&g(y_{k+1}) \\
=&g(y_k) + \int_0^1 \langle y_{k+1} - y_k, \nabla g(y_k + t(y_{k+1} - y_k)) \rangle dt \\
=&g(y_k) + \langle \nabla g(y_k), y_{k+1} - y_k \rangle + \int_0^1 \big\langle y_{k+1} - y_k, \\
&\qquad \nabla g(y_k + t(y_{k+1} - y_k)) - \nabla g(y_k) \big\rangle dt \\
=&g(y_k) + \left\langle \nabla_{i_k} g(y_k), y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle + \int_0^1 \big\langle y_{k+1}^{(i_k)} - y_k^{(i_k)}, \\
&\nabla_{i_k} g(y_k + t(y_{k+1} - y_k)) - \nabla_{i_k} g(y_k) \big\rangle dt \\
\leq&g(y_k) + \left\langle \nabla_{i_k} g(y_k), y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle + \int_0^1 \left\| y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\| \\
&\|\nabla_{i_k} g(y_k + t(y_{k+1} - y_k)) - \nabla_{i_k} g(y_k)\| \, dt \\
\leq&g(y_k) + \left\langle \nabla_{i_k} g(y_k), y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle + \\
&\int_0^1 L_{i_k} \left\| y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\|^2 t dt \\
=&g(y_k) + \left\langle \nabla_{i_k} g(y_k), y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle + \frac{L_{i_k}}{2} \left\| y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\|^2 .
\end{aligned}$$

On the third line, we have used the fact that y_k and y_{k+1} agree on all coordinate blocks except the i_k-th block. On the fourth line, we have used the Cauchy-Schwartz inequality. On the fifth line, we have used the fact that $\nabla_{i_k} g(\cdot)$ is L_{i_k}-Lipschitz. □

Proof of Equation (4): We have

$$\begin{aligned}
&\|y_{k+1} - y^*\|^2 \\
&= \|y_k - y^*\|^2 + \|y_{k+1} - y_k\|^2 + 2\langle y_k - y^*, y_{k+1} - y_k \rangle \\
&= \|y_k - y^*\|^2 - \|y_{k+1} - y_k\|^2 + 2 \langle y_{k+1} - y^*, y_{k+1} - y_k \rangle
\end{aligned}$$

$$
\begin{aligned}
&= \|y_k - y^*\|^2 - \left\|y_{k+1}^{(i_k)} - y_k^{(i_k)}\right\|^2 + \\
&\quad 2\left\langle y_{k+1}^{(i_k)} - (y^*)^{(i_k)}, y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle \\
&\overset{(2)}{\leq} \|y_k - y^*\|^2 - \left\|y_{k+1}^{(i_k)} - y_k^{(i_k)}\right\|^2 + \\
&\quad \frac{2}{L_{i_k}}\left\langle \nabla_{i_k} g(y_k), (y^*)^{(i_k)} - y_{k+1}^{(i_k)} \right\rangle \\
&= \|y_k - y^*\|^2 + \frac{2}{L_{i_k}}\left\langle \nabla_{i_k} g(y_k), (y^*)^{(i_k)} - y_k^{(i_k)} \right\rangle \\
&- \frac{2}{L_{i_k}}\left(\frac{L_{i_k}}{2}\left\|y_{k+1}^{(i_k)} - y_k^{(i_k)}\right\|^2 + \left\langle \nabla_{i_k} g(y_k), y_{k+1}^{(i_k)} - y_k^{(i_k)} \right\rangle \right) \\
&\overset{(3)}{\leq} \|y_k - y^*\|^2 + \frac{2}{L_{i_k}}\left\langle \nabla_{i_k} g(y_k), (y^*)^{(i_k)} - y_k^{(i_k)} \right\rangle - \\
&\quad \frac{2}{L_{i_k}}\left(g(y_{k+1}) - g(y_k)\right). \qquad (10)
\end{aligned}
$$

On the third line, we have used the fact that y_k and y_{k+1} agree on all coordinate blocks except the i_k-th block. On the fourth line, we have used the inequality (2) with $z = (y^*)^{(i_k)}$. On the last line, we have used inequality (3). $\square$