2

Precursors

In this section, we briefly review two optimization algorithms that are precursors to the alternating direction method of multipliers. While we will not use this material in the sequel, it provides some useful background and motivation.

2.1 Dual Ascent

Consider the equality-constrained convex optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b, \end{array} \tag{2.1}$$

with variable $x \in \mathbf{R}^n$, where $A \in \mathbf{R}^{m \times n}$ and $f : \mathbf{R}^n \to \mathbf{R}$ is convex.

The Lagrangian for problem (2.1) is

$$L(x, y) = f(x) + y^T(Ax - b)$$

and the dual function is

$$g(y) = \inf_x L(x, y) = -f^*(-A^T y) - b^T y,$$

where y is the dual variable or Lagrange multiplier, and f^* is the convex conjugate of f; see [20, §3.3] or [140, §12] for background. The dual

problem is

$$\text{maximize} \quad g(y),$$

with variable $y \in \mathbf{R}^m$. Assuming that strong duality holds, the optimal values of the primal and dual problems are the same. We can recover a primal optimal point $x^\star$ from a dual optimal point $y^\star$ as

$$x^\star = \operatorname*{argmin}_x L(x, y^\star),$$

provided there is only one minimizer of $L(x, y^\star)$. (This is the case if, *e.g.*, f is strictly convex.) In the sequel, we will use the notation $\operatorname{argmin}_x F(x)$ to denote *any* minimizer of F, even when F does not have a unique minimizer.

In the *dual ascent method*, we solve the dual problem using gradient ascent. Assuming that g is differentiable, the gradient $\nabla g(y)$ can be evaluated as follows. We first find $x^+ = \operatorname{argmin}_x L(x, y)$; then we have $\nabla g(y) = Ax^+ - b$, which is the residual for the equality constraint. The dual ascent method consists of iterating the updates

$$x^{k+1} := \operatorname*{argmin}_x L(x, y^k) \tag{2.2}$$

$$y^{k+1} := y^k + \alpha^k (Ax^{k+1} - b), \tag{2.3}$$

where $\alpha^k > 0$ is a step size, and the superscript is the iteration counter. The first step (2.2) is an x-minimization step, and the second step (2.3) is a dual variable update. The dual variable y can be interpreted as a vector of prices, and the y-update is then called a *price update* or *price adjustment* step. This algorithm is called dual ascent since, with appropriate choice of α^k, the dual function increases in each step, *i.e.*, $g(y^{k+1}) > g(y^k)$.

The dual ascent method can be used even in some cases when g is not differentiable. In this case, the residual $Ax^{k+1} - b$ is not the gradient of g, but the negative of a *subgradient* of $-g$. This case requires a different choice of the α^k than when g is differentiable, and convergence is not monotone; it is often the case that $g(y^{k+1}) \not> g(y^k)$. In this case, the algorithm is usually called the *dual subgradient method* [152].

If α^k is chosen appropriately and several other assumptions hold, then x^k converges to an optimal point and y^k converges to an optimal

dual point. However, these assumptions do not hold in many applications, so dual ascent often cannot be used. As an example, if f is a nonzero affine function of any component of x, then the x-update (2.2) fails, since L is unbounded below in x for most y.

2.2 Dual Decomposition

The major benefit of the dual ascent method is that it can lead to a decentralized algorithm in some cases. Suppose, for example, that the objective f is *separable* (with respect to a partition or splitting of the variable into subvectors), meaning that

$$f(x) = \sum_{i=1}^{N} f_i(x_i),$$

where $x = (x_1, \ldots, x_N)$ and the variables $x_i \in \mathbf{R}^{n_i}$ are subvectors of x. Partitioning the matrix A conformably as

$$A = [A_1 \ \cdots \ A_N],$$

so $Ax = \sum_{i=1}^{N} A_i x_i$, the Lagrangian can be written as

$$L(x, y) = \sum_{i=1}^{N} L_i(x_i, y) = \sum_{i=1}^{N} \left(f_i(x_i) + y^T A_i x_i - (1/N) y^T b \right),$$

which is also separable in x. This means that the x-minimization step (2.2) splits into N separate problems that can be solved in parallel. Explicitly, the algorithm is

$$x_i^{k+1} := \operatorname*{argmin}_{x_i} L_i(x_i, y^k) \tag{2.4}$$

$$y^{k+1} := y^k + \alpha^k (A x^{k+1} - b). \tag{2.5}$$

The x-minimization step (2.4) is carried out independently, in parallel, for each $i = 1, \ldots, N$. In this case, we refer to the dual ascent method as *dual decomposition*.

In the general case, each iteration of the dual decomposition method requires a *broadcast* and a *gather* operation. In the dual update step (2.5), the equality constraint residual contributions $A_i x_i^{k+1}$ are

collected (gathered) in order to compute the residual $Ax^{k+1} - b$. Once the (global) dual variable y^{k+1} is computed, it must be distributed (broadcast) to the processors that carry out the N individual x_i minimization steps (2.4).

Dual decomposition is an old idea in optimization, and traces back at least to the early 1960s. Related ideas appear in well known work by Dantzig and Wolfe [44] and Benders [13] on large-scale linear programming, as well as in Dantzig's seminal book [43]. The general idea of dual decomposition appears to be originally due to Everett [69], and is explored in many early references [107, 84, 117, 14]. The use of nondifferentiable optimization, such as the subgradient method, to solve the dual problem is discussed by Shor [152]. Good references on dual methods and decomposition include the book by Bertsekas [16, chapter 6] and the survey by Nedić and Ozdaglar [131] on distributed optimization, which discusses dual decomposition methods and consensus problems. A number of papers also discuss variants on standard dual decomposition, such as [129].

More generally, decentralized optimization has been an active topic of research since the 1980s. For instance, Tsitsiklis and his co-authors worked on a number of decentralized detection and consensus problems involving the minimization of a smooth function f known to multiple agents [160, 161, 17]. Some good reference books on parallel optimization include those by Bertsekas and Tsitsiklis [17] and Censor and Zenios [31]. There has also been some recent work on problems where each agent has its own convex, potentially nondifferentiable, objective function [130]. See [54] for a recent discussion of distributed methods for graph-structured optimization problems.

2.3 Augmented Lagrangians and the Method of Multipliers

Augmented Lagrangian methods were developed in part to bring robustness to the dual ascent method, and in particular, to yield convergence without assumptions like strict convexity or finiteness of f. The *augmented Lagrangian* for (2.1) is

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + (\rho/2)\|Ax - b\|_2^2, \tag{2.6}$$

where $\rho > 0$ is called the *penalty parameter*. (Note that L_0 is the standard Lagrangian for the problem.) The augmented Lagrangian can be viewed as the (unaugmented) Lagrangian associated with the problem

$$\begin{array}{ll} \text{minimize} & f(x) + (\rho/2)\|Ax - b\|_2^2 \\ \text{subject to} & Ax = b. \end{array}$$

This problem is clearly equivalent to the original problem (2.1), since for any feasible x the term added to the objective is zero. The associated dual function is $g_\rho(y) = \inf_x L_\rho(x, y)$.

The benefit of including the penalty term is that g_ρ can be shown to be differentiable under rather mild conditions on the original problem. The gradient of the augmented dual function is found the same way as with the ordinary Lagrangian, *i.e.*, by minimizing over x, and then evaluating the resulting equality constraint residual. Applying dual ascent to the modified problem yields the algorithm

$$x^{k+1} := \operatorname*{argmin}_x L_\rho(x, y^k) \tag{2.7}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} - b), \tag{2.8}$$

which is known as the *method of multipliers* for solving (2.1). This is the same as standard dual ascent, except that the x-minimization step uses the augmented Lagrangian, and the penalty parameter ρ is used as the step size α^k. The method of multipliers converges under far more general conditions than dual ascent, including cases when f takes on the value $+\infty$ or is not strictly convex.

It is easy to motivate the choice of the particular step size ρ in the dual update (2.8). For simplicity, we assume here that f is differentiable, though this is not required for the algorithm to work. The optimality conditions for (2.1) are primal and dual feasibility, *i.e.*,

$$Ax^\star - b = 0, \qquad \nabla f(x^\star) + A^T y^\star = 0,$$

respectively. By definition, x^{k+1} minimizes $L_\rho(x, y^k)$, so

$$\begin{aligned} 0 &= \nabla_x L_\rho(x^{k+1}, y^k) \\ &= \nabla_x f(x^{k+1}) + A^T\left(y^k + \rho(Ax^{k+1} - b)\right) \\ &= \nabla_x f(x^{k+1}) + A^T y^{k+1}. \end{aligned}$$

We see that by using ρ as the step size in the dual update, the iterate (x^{k+1}, y^{k+1}) is dual feasible. As the method of multipliers proceeds, the primal residual $Ax^{k+1} - b$ converges to zero, yielding optimality.

The greatly improved convergence properties of the method of multipliers over dual ascent comes at a cost. When f is separable, the augmented Lagrangian L_ρ is not separable, so the x-minimization step (2.7) cannot be carried out separately in parallel for each x_i. This means that the basic method of multipliers cannot be used for decomposition. We will see how to address this issue next.

Augmented Lagrangians and the method of multipliers for constrained optimization were first proposed in the late 1960s by Hestenes [97, 98] and Powell [138]. Many of the early numerical experiments on the method of multipliers are due to Miele et al. [124, 125, 126]. Much of the early work is consolidated in a monograph by Bertsekas [15], who also discusses similarities to older approaches using Lagrangians and penalty functions [6, 5, 71], as well as a number of generalizations.

3

Alternating Direction Method of Multipliers

3.1 Algorithm

ADMM is an algorithm that is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers. The algorithm solves problems in the form

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array} \tag{3.1}$$

with variables $x \in \mathbf{R}^n$ and $z \in \mathbf{R}^m$, where $A \in \mathbf{R}^{p\times n}$, $B \in \mathbf{R}^{p\times m}$, and $c \in \mathbf{R}^p$. We will assume that f and g are convex; more specific assumptions will be discussed in §3.2. The only difference from the general linear equality-constrained problem (2.1) is that the variable, called x there, has been split into two parts, called x and z here, with the objective function separable across this splitting. The optimal value of the problem (3.1) will be denoted by

$$p^\star = \inf\{f(x) + g(z) \mid Ax + Bz = c\}.$$

As in the method of multipliers, we form the augmented Lagrangian

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2.$$

ADMM consists of the iterations

$$x^{k+1} := \operatorname*{argmin}_x L_\rho(x, z^k, y^k) \tag{3.2}$$

$$z^{k+1} := \operatorname*{argmin}_z L_\rho(x^{k+1}, z, y^k) \tag{3.3}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \tag{3.4}$$

where $\rho > 0$. The algorithm is very similar to dual ascent and the method of multipliers: it consists of an x-minimization step (3.2), a z-minimization step (3.3), and a dual variable update (3.4). As in the method of multipliers, the dual variable update uses a step size equal to the augmented Lagrangian parameter ρ.

The method of multipliers for (3.1) has the form

$$\begin{aligned}(x^{k+1}, z^{k+1}) &:= \operatorname*{argmin}_{x,z} L_\rho(x, z, y^k)\\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c).\end{aligned}$$

Here the augmented Lagrangian is minimized jointly with respect to the two primal variables. In ADMM, on the other hand, x and z are updated in an alternating or sequential fashion, which accounts for the term *alternating direction*. ADMM can be viewed as a version of the method of multipliers where a single *Gauss-Seidel* pass [90, §10.1] over x and z is used instead of the usual joint minimization. Separating the minimization over x and z into two steps is precisely what allows for decomposition when f or g are separable.

The algorithm state in ADMM consists of z^k and y^k. In other words, (z^{k+1}, y^{k+1}) is a function of (z^k, y^k). The variable x^k is not part of the state; it is an intermediate result computed from the previous state (z^{k-1}, y^{k-1}).

If we switch (re-label) x and z, f and g, and A and B in the problem (3.1), we obtain a variation on ADMM with the order of the x-update step (3.2) and z-update step (3.3) reversed. The roles of x and z are almost symmetric, but not quite, since the dual update is done after the z-update but before the x-update.

3.1.1 Scaled Form

ADMM can be written in a slightly different form, which is often more convenient, by combining the linear and quadratic terms in the augmented Lagrangian and scaling the dual variable. Defining the residual $r = Ax + Bz - c$, we have

$$\begin{aligned} y^T r + (\rho/2)\|r\|_2^2 &= (\rho/2)\|r + (1/\rho)y\|_2^2 - (1/2\rho)\|y\|_2^2 \\ &= (\rho/2)\|r + u\|_2^2 - (\rho/2)\|u\|_2^2, \end{aligned}$$

where $u = (1/\rho)y$ is the *scaled dual variable*. Using the scaled dual variable, we can express ADMM as

$$x^{k+1} := \operatorname*{argmin}_x \left(f(x) + (\rho/2)\|Ax + Bz^k - c + u^k\|_2^2 \right) \tag{3.5}$$

$$z^{k+1} := \operatorname*{argmin}_z \left(g(z) + (\rho/2)\|Ax^{k+1} + Bz - c + u^k\|_2^2 \right) \tag{3.6}$$

$$u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c. \tag{3.7}$$

Defining the residual at iteration k as $r^k = Ax^k + Bz^k - c$, we see that

$$u^k = u^0 + \sum_{j=1}^{k} r^j,$$

the running sum of the residuals.

We call the first form of ADMM above, given by (3.2–3.4), the *unscaled form*, and the second form (3.5–3.7) the *scaled form*, since it is expressed in terms of a scaled version of the dual variable. The two are clearly equivalent, but the formulas in the scaled form of ADMM are often shorter than in the unscaled form, so we will use the scaled form in the sequel. We will use the unscaled form when we wish to emphasize the role of the dual variable or to give an interpretation that relies on the (unscaled) dual variable.

3.2 Convergence

There are many convergence results for ADMM discussed in the literature; here, we limit ourselves to a basic but still very general result that applies to all of the examples we will consider. We will make one

assumption about the functions f and g, and one assumption about problem (3.1).

Assumption 1. The (extended-real-valued) functions $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ and $g : \mathbf{R}^m \to \mathbf{R} \cup \{+\infty\}$ are closed, proper, and convex.

This assumption can be expressed compactly using the epigraphs of the functions: The function f satisfies assumption 1 if and only if its epigraph

$$\mathbf{epi}\, f = \{(x,t) \in \mathbf{R}^n \times \mathbf{R} \mid f(x) \leq t\}$$

is a closed nonempty convex set.

Assumption 1 implies that the subproblems arising in the x-update (3.2) and z-update (3.3) are *solvable*, *i.e.*, there exist x and z, not necessarily unique (without further assumptions on A and B), that minimize the augmented Lagrangian. It is important to note that assumption 1 allows f and g to be nondifferentiable and to assume the value $+\infty$. For example, we can take f to be the indicator function of a closed nonempty convex set $\mathcal{C}$, *i.e.*, $f(x) = 0$ for $x \in \mathcal{C}$ and $f(x) = +\infty$ otherwise. In this case, the x-minimization step (3.2) will involve solving a constrained quadratic program over $\mathcal{C}$, the effective domain of f.

Assumption 2. The unaugmented Lagrangian L_0 has a saddle point.

Explicitly, there exist $(x^\star, z^\star, y^\star)$, not necessarily unique, for which

$$L_0(x^\star, z^\star, y) \leq L_0(x^\star, z^\star, y^\star) \leq L_0(x, z, y^\star)$$

holds for all $x,\ z,\ y$.

By assumption 1, it follows that $L_0(x^\star, z^\star, y^\star)$ is finite for any saddle point $(x^\star, z^\star, y^\star)$. This implies that $(x^\star, z^\star)$ is a solution to (3.1), so $Ax^\star + Bz^\star = c$ and $f(x^\star) < \infty$, $g(z^\star) < \infty$. It also implies that $y^\star$ is dual optimal, and the optimal values of the primal and dual problems are equal, *i.e.*, that strong duality holds. Note that we make no assumptions about A, B, or c, except implicitly through assumption 2; in particular, neither A nor B is required to be full rank.

3.2.1 Convergence

Under assumptions 1 and 2, the ADMM iterates satisfy the following:

- *Residual convergence.* $r^k \to 0$ as $k \to \infty$, *i.e.*, the iterates approach feasibility.
- *Objective convergence.* $f(x^k) + g(z^k) \to p^\star$ as $k \to \infty$, *i.e.*, the objective function of the iterates approaches the optimal value.
- *Dual variable convergence.* $y^k \to y^\star$ as $k \to \infty$, where $y^\star$ is a dual optimal point.

A proof of the residual and objective convergence results is given in appendix A. Note that x^k and z^k need not converge to optimal values, although such results can be shown under additional assumptions.

3.2.2 Convergence in Practice

Simple examples show that ADMM can be very slow to converge to high accuracy. However, it is often the case that ADMM converges to modest accuracy—sufficient for many applications—within a few tens of iterations. This behavior makes ADMM similar to algorithms like the conjugate gradient method, for example, in that a few tens of iterations will often produce acceptable results of practical use. However, the slow convergence of ADMM also distinguishes it from algorithms such as Newton's method (or, for constrained problems, interior-point methods), where high accuracy can be attained in a reasonable amount of time. While in some cases it is possible to combine ADMM with a method for producing a high accuracy solution from a low accuracy solution [64], in the general case ADMM will be practically useful mostly in cases when modest accuracy is sufficient. Fortunately, this is usually the case for the kinds of large-scale problems we consider. Also, in the case of statistical and machine learning problems, solving a parameter estimation problem to very high accuracy often yields little to no improvement in actual prediction performance, the real metric of interest in applications.

3.3 Optimality Conditions and Stopping Criterion

The necessary and sufficient optimality conditions for the ADMM problem (3.1) are primal feasibility,

$$Ax^\star + Bz^\star - c = 0, \tag{3.8}$$

and dual feasibility,

$$0 \in \partial f(x^\star) + A^T y^\star \tag{3.9}$$

$$0 \in \partial g(z^\star) + B^T y^\star. \tag{3.10}$$

Here, ∂ denotes the subdifferential operator; see, *e.g.*, [140, 19, 99]. (When f and g are differentiable, the subdifferentials ∂f and ∂g can be replaced by the gradients ∇f and ∇g, and $\in$ can be replaced by $=$.)

Since z^{k+1} minimizes $L_\rho(x^{k+1}, z, y^k)$ by definition, we have that

$$\begin{aligned} 0 &\in \partial g(z^{k+1}) + B^T y^k + \rho B^T (Ax^{k+1} + Bz^{k+1} - c) \\ &= \partial g(z^{k+1}) + B^T y^k + \rho B^T r^{k+1} \\ &= \partial g(z^{k+1}) + B^T y^{k+1}. \end{aligned}$$

This means that z^{k+1} and y^{k+1} always satisfy (3.10), so attaining optimality comes down to satisfying (3.8) and (3.9). This phenomenon is analogous to the iterates of the method of multipliers always being dual feasible; see page 11.

Since x^{k+1} minimizes $L_\rho(x, z^k, y^k)$ by definition, we have that

$$\begin{aligned} 0 &\in \partial f(x^{k+1}) + A^T y^k + \rho A^T (Ax^{k+1} + Bz^k - c) \\ &= \partial f(x^{k+1}) + A^T (y^k + \rho r^{k+1} + \rho B(z^k - z^{k+1})) \\ &= \partial f(x^{k+1}) + A^T y^{k+1} + \rho A^T B(z^k - z^{k+1}), \end{aligned}$$

or equivalently,

$$\rho A^T B(z^{k+1} - z^k) \in \partial f(x^{k+1}) + A^T y^{k+1}.$$

This means that the quantity

$$s^{k+1} = \rho A^T B(z^{k+1} - z^k)$$

can be viewed as a residual for the dual feasibility condition (3.9). We will refer to s^{k+1} as the *dual residual* at iteration $k+1$, and to $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ as the *primal residual* at iteration $k+1$.

In summary, the optimality conditions for the ADMM problem consist of three conditions, (3.8–3.10). The last condition (3.10) always holds for $(x^{k+1}, z^{k+1}, y^{k+1})$; the residuals for the other two, (3.8) and (3.9), are the primal and dual residuals r^{k+1} and s^{k+1}, respectively. These two residuals converge to zero as ADMM proceeds. (In fact, the convergence proof in appendix A shows $B(z^{k+1} - z^k)$ converges to zero, which implies s^k converges to zero.)

3.3.1 Stopping Criteria

The residuals of the optimality conditions can be related to a bound on the objective suboptimality of the current point, *i.e.*, $f(x^k) + g(z^k) - p^\star$. As shown in the convergence proof in appendix A, we have

$$f(x^k) + g(z^k) - p^\star \leq -(y^k)^T r^k + (x^k - x^\star)^T s^k. \tag{3.11}$$

This shows that when the residuals r^k and s^k are small, the objective suboptimality also must be small. We cannot use this inequality directly in a stopping criterion, however, since we do not know $x^\star$. But if we guess or estimate that $\|x^k - x^\star\|_2 \leq d$, we have that

$$f(x^k) + g(z^k) - p^\star \leq -(y^k)^T r^k + d\|s^k\|_2 \leq \|y^k\|_2\|r^k\|_2 + d\|s^k\|_2.$$

The middle or righthand terms can be used as an approximate bound on the objective suboptimality (which depends on our guess of d).

This suggests that a reasonable termination criterion is that the primal and dual residuals must be small, *i.e.*,

$$\|r^k\|_2 \leq \epsilon^{\text{pri}} \quad \text{and} \quad \|s^k\|_2 \leq \epsilon^{\text{dual}}, \tag{3.12}$$

where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are feasibility tolerances for the primal and dual feasibility conditions (3.8) and (3.9), respectively. These tolerances can be chosen using an absolute and relative criterion, such as

$$\begin{aligned} \epsilon^{\text{pri}} &= \sqrt{p}\,\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\{\|Ax^k\|_2, \|Bz^k\|_2, \|c\|_2\}, \\ \epsilon^{\text{dual}} &= \sqrt{n}\,\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|A^T y^k\|_2, \end{aligned}$$

where $\epsilon^{\text{abs}} > 0$ is an absolute tolerance and $\epsilon^{\text{rel}} > 0$ is a relative tolerance. (The factors $\sqrt{p}$ and $\sqrt{n}$ account for the fact that the ℓ_2 norms are in $\mathbf{R}^p$ and $\mathbf{R}^n$, respectively.) A reasonable value for the relative stopping

criterion might be $\epsilon^{\text{rel}} = 10^{-3}$ or 10^{-4}, depending on the application. The choice of absolute stopping criterion depends on the scale of the typical variable values.

3.4 Extensions and Variations

Many variations on the classic ADMM algorithm have been explored in the literature. Here we briefly survey some of these variants, organized into groups of related ideas. Some of these methods can give superior convergence in practice compared to the standard ADMM presented above. Most of the extensions have been rigorously analyzed, so the convergence results described above are still valid (in some cases, under some additional conditions).

3.4.1 Varying Penalty Parameter

A standard extension is to use possibly different penalty parameters ρ^k for each iteration, with the goal of improving the convergence in practice, as well as making performance less dependent on the initial choice of the penalty parameter. In the context of the method of multipliers, this approach is analyzed in [142], where it is shown that superlinear convergence may be achieved if $\rho^k \to \infty$. Though it can be difficult to prove the convergence of ADMM when ρ varies by iteration, the fixed-ρ theory still applies if one just assumes that ρ becomes fixed after a finite number of iterations.

A simple scheme that often works well is (see, *e.g.*, [96, 169]):

$$\rho^{k+1} := \begin{cases} \tau^{\text{incr}}\rho^k & \text{if } \|r^k\|_2 > \mu\|s^k\|_2 \\ \rho^k/\tau^{\text{decr}} & \text{if } \|s^k\|_2 > \mu\|r^k\|_2 \\ \rho^k & \text{otherwise,} \end{cases} \tag{3.13}$$

where $\mu > 1$, $\tau^{\text{incr}} > 1$, and $\tau^{\text{decr}} > 1$ are parameters. Typical choices might be $\mu = 10$ and $\tau^{\text{incr}} = \tau^{\text{decr}} = 2$. The idea behind this penalty parameter update is to try to keep the primal and dual residual norms within a factor of μ of one another as they both converge to zero.

The ADMM update equations suggest that large values of ρ place a large penalty on violations of primal feasibility and so tend to produce

small primal residuals. Conversely, the definition of s^{k+1} suggests that small values of ρ tend to reduce the dual residual, but at the expense of reducing the penalty on primal feasibility, which may result in a larger primal residual. The adjustment scheme (3.13) inflates ρ by τ^{incr} when the primal residual appears large compared to the dual residual, and deflates ρ by τ^{decr} when the primal residual seems too small relative to the dual residual. This scheme may also be refined by taking into account the relative magnitudes of ϵ^{pri} and ϵ^{dual}.

When a varying penalty parameter is used in the scaled form of ADMM, the scaled dual variable $u^k = (1/\rho)y^k$ must also be rescaled after updating ρ; for example, if ρ is halved, u^k should be doubled before proceeding.

3.4.2 More General Augmenting Terms

Another idea is to allow for a different penalty parameter for each constraint, or more generally, to replace the quadratic term $(\rho/2)\|r\|_2^2$ with $(1/2)r^T Pr$, where P is a symmetric positive definite matrix. When P is constant, we can interpret this generalized version of ADMM as standard ADMM applied to a modified initial problem with the equality constraints $r = 0$ replaced with $Fr = 0$, where $F^T F = P$.

3.4.3 Over-relaxation

In the z- and y-updates, the quantity Ax^{k+1} can be replaced with

$$\alpha^k Ax^{k+1} - (1 - \alpha^k)(Bz^k - c),$$

where $\alpha^k \in (0, 2)$ is a *relaxation parameter*; when $\alpha^k > 1$, this technique is called *over-relaxation*, and when $\alpha^k < 1$, it is called *under-relaxation*. This scheme is analyzed in [63], and experiments in [59, 64] suggest that over-relaxation with $\alpha^k \in [1.5, 1.8]$ can improve convergence.

3.4.4 Inexact Minimization

ADMM will converge even when the x- and z-minimization steps are not carried out exactly, provided certain suboptimality measures

in the minimizations satisfy an appropriate condition, such as being summable. This result is due to Eckstein and Bertsekas [63], building on earlier results by Gol'shtein and Tret'yakov [89]. This technique is important in situations where the x- or z-updates are carried out using an iterative method; it allows us to solve the minimizations only approximately at first, and then more accurately as the iterations progress.

3.4.5 Update Ordering

Several variations on ADMM involve performing the x-, z-, and y-updates in varying orders or multiple times. For example, we can divide the variables into k blocks, and update each of them in turn, possibly multiple times, before performing each dual variable update; see, *e.g.*, [146]. Carrying out multiple x- and z-updates before the y-update can be interpreted as executing multiple Gauss-Seidel passes instead of just one; if many sweeps are carried out before each dual update, the resulting algorithm is very close to the standard method of multipliers [17, §3.4.4]. Another variation is to perform an additional y-update between the x- and z-update, with half the step length [17].

3.4.6 Related Algorithms

There are also a number of other algorithms distinct from but inspired by ADMM. For instance, Fukushima [80] applies ADMM to a dual problem formulation, yielding a 'dual ADMM' algorithm, which is shown in [65] to be equivalent to the 'primal Douglas-Rachford' method discussed in [57, §3.5.6]. As another example, Zhu et al. [183] discuss variations of distributed ADMM (discussed in §7, §8, and §10) that can cope with various complicating factors, such as noise in the messages exchanged for the updates, or asynchronous updates, which can be useful in a regime when some processors or subsystems randomly fail. There are also algorithms resembling a combination of ADMM and the *proximal* method of multipliers [141], rather than the standard method of multipliers; see, *e.g.*, [33, 60]. Other representative publications include [62, 143, 59, 147, 158, 159, 42].

3.5 Notes and References

ADMM was originally proposed in the mid-1970s by Glowinski and Marrocco [86] and Gabay and Mercier [82]. There are a number of other important papers analyzing the properties of the algorithm, including [76, 81, 75, 87, 157, 80, 65, 33]. In particular, the convergence of ADMM has been explored by many authors, including Gabay [81] and Eckstein and Bertsekas [63].

ADMM has also been applied to a number of statistical problems, such as constrained sparse regression [18], sparse signal recovery [70], image restoration and denoising [72, 154, 134], trace norm regularized least squares minimization [174], sparse inverse covariance selection [178], the Dantzig selector [116], and support vector machines [74], among others. For examples in signal processing, see [42, 40, 41, 150, 149] and the references therein.

Many papers analyzing ADMM do so from the perspective of *maximal monotone operators* [23, 141, 142, 63, 144]. Briefly, a wide variety of problems can be posed as finding a zero of a maximal monotone operator; for example, if f is closed, proper, and convex, then the subdifferential operator ∂f is maximal monotone, and finding a zero of ∂f is simply minimization of f; such a minimization may implicitly contain constraints if f is allowed to take the value $+\infty$. Rockafellar's *proximal point algorithm* [142] is a general method for finding a zero of a maximal monotone operator, and a wide variety of algorithms have been shown to be special cases, including proximal minimization (see §4.1), the method of multipliers, and ADMM. For a more detailed review of the older literature, see [57, §2].

The method of multipliers was shown to be a special case of the proximal point algorithm by Rockafellar [141]. Gabay [81] showed that ADMM is a special case of a method called *Douglas-Rachford splitting* for monotone operators [53, 112], and Eckstein and Bertsekas [63] showed in turn that Douglas-Rachford splitting is a special case of the proximal point algorithm. (The variant of ADMM that performs an extra y-update between the x- and z-updates is equivalent to *Peaceman-Rachford splitting* [137, 112] instead, as shown by Glowinski and Le Tallec [87].) Using the same framework, Eckstein

and Bertsekas [63] also showed the relationships between a number of other algorithms, such as Spingarn's method of partial inverses [153]. Lawrence and Spingarn [108] develop an alternative framework showing that Douglas-Rachford splitting, hence ADMM, is a special case of the proximal point algorithm; Eckstein and Ferris [64] offer a more recent discussion explaining this approach.

The major importance of these results is that they allow the powerful convergence theory for the proximal point algorithm to apply directly to ADMM and other methods, and show that many of these algorithms are essentially identical. (But note that our proof of convergence of the basic ADMM algorithm, given in appendix A, is self-contained and does not rely on this abstract machinery.) Research on operator splitting methods and their relation to decomposition algorithms continues to this day [66, 67].

A considerable body of recent research considers replacing the quadratic penalty term in the standard method of multipliers with a more general deviation penalty, such as one derived from a *Bregman divergence* [30, 58]; see [22] for background material. Unfortunately, these generalizations do not appear to carry over in a straightforward manner from non-decomposition augmented Lagrangian methods to ADMM: There is currently no proof of convergence known for ADMM with nonquadratic penalty terms.

12

Conclusions

We have discussed ADMM and illustrated its applicability to distributed convex optimization in general and many problems in statistical machine learning in particular. We argue that ADMM can serve as a good general-purpose tool for optimization problems arising in the analysis and processing of modern massive datasets. Much like gradient descent and the conjugate gradient method are standard tools of great use when optimizing smooth functions on a single machine, ADMM should be viewed as an analogous tool in the distributed regime.

ADMM sits at a higher level of abstraction than classical optimization algorithms like Newton's method. In such algorithms, the base operations are low-level, consisting of linear algebra operations and the computation of gradients and Hessians. In the case of ADMM, the base operations include solving small convex optimization problems (which in some cases can be done via a simple analytical formula). For example, when applying ADMM to a very large model fitting problem, each update reduces to a (regularized) model fitting problem on a smaller dataset. These subproblems can be solved using any standard serial algorithm suitable for small to medium sized problems. In this sense, ADMM builds on existing algorithms for single machines, and so can be

viewed as a modular coordination algorithm that 'incentivizes' a set of simpler algorithms to collaborate to solve much larger global problems together than they could on their own. Alternatively, it can be viewed as a simple way of 'bootstrapping' specialized algorithms for small to medium sized problems to work on much larger problems than would otherwise be possible.

We emphasize that for any particular problem, it is likely that another method will perform better than ADMM, or that some variation on ADMM will substantially improve performance. However, a simple algorithm derived from basic ADMM will often offer performance that is at least comparable to very specialized algorithms (even in the serial setting), and in most cases, the simple ADMM algorithm will be efficient enough to be useful. In a few cases, ADMM-based methods actually turn out to be state-of-the-art even in the serial regime. Moreover, ADMM has the benefit of being extremely simple to implement, and it maps onto several standard distributed programming models reasonably well.

ADMM was developed over a generation ago, with its roots stretching far in advance of the Internet, distributed and cloud computing systems, massive high-dimensional datasets, and the associated large-scale applied statistical problems. Despite this, it appears to be well suited to the modern regime, and has the important benefit of being quite general in its scope and applicability.

A

Convergence Proof

The basic convergence result given in §3.2 can be found in several references, such as [81, 63]. Many of these give more sophisticated results, with more general penalties or inexact minimization. For completeness, we give a proof here.

We will show that if f and g are closed, proper, and convex, and the Lagrangian L_0 has a saddle point, then we have primal residual convergence, meaning that $r^k \to 0$, and objective convergence, meaning that $p^k \to p^\star$, where $p^k = f(x^k) + g(z^k)$. We will also see that the dual residual $s^k = \rho A^T B(z^k - z^{k-1})$ converges to zero.

Let $(x^\star, z^\star, y^\star)$ be a saddle point for L_0, and define

$$V^k = (1/\rho)\|y^k - y^\star\|_2^2 + \rho\|B(z^k - z^\star)\|_2^2,$$

We will see that V^k is a *Lyapunov function* for the algorithm, *i.e.*, a nonnegative quantity that decreases in each iteration. (Note that V^k is unknown while the algorithm runs, since it depends on the unknown values $z^\star$ and $y^\star$.)

We first outline the main idea. The proof relies on three key inequalities, which we will prove below using basic results from convex analysis

along with simple algebra. The first inequality is

$$V^{k+1} \leq V^k - \rho\|r^{k+1}\|_2^2 - \rho\|B(z^{k+1} - z^k)\|_2^2. \tag{A.1}$$

This states that V^k decreases in each iteration by an amount that depends on the norm of the residual and on the change in z over one iteration. Because $V^k \leq V^0$, it follows that y^k and Bz^k are bounded. Iterating the inequality above gives that

$$\rho\sum_{k=0}^{\infty}\left(\|r^{k+1}\|_2^2 + \|B(z^{k+1} - z^k)\|_2^2\right) \leq V^0,$$

which implies that $r^k \to 0$ and $B(z^{k+1} - z^k) \to 0$ as $k \to \infty$. Multiplying the second expression by ρA^T shows that the dual residual $s^k = \rho A^T B(z^{k+1} - z^k)$ converges to zero. (This shows that the stopping criterion (3.12), which requires the primal and dual residuals to be small, will eventually hold.)

The second key inequality is

$$\begin{aligned} &p^{k+1} - p^\star \\ &\quad \leq -(y^{k+1})^T r^{k+1} - \rho(B(z^{k+1} - z^k))^T(-r^{k+1} + B(z^{k+1} - z^\star)), \end{aligned} \tag{A.2}$$

and the third inequality is

$$p^\star - p^{k+1} \leq y^{\star T} r^{k+1}. \tag{A.3}$$

The righthand side in (A.2) goes to zero as $k \to \infty$, because $B(z^{k+1} - z^\star)$ is bounded and both r^{k+1} and $B(z^{k+1} - z^k)$ go to zero. The righthand side in (A.3) goes to zero as $k \to \infty$, since r^k goes to zero. Thus we have $\lim_{k\to\infty} p^k = p^\star$, *i.e.*, objective convergence.

Before giving the proofs of the three key inequalities, we derive the inequality (3.11) mentioned in our discussion of stopping criterion from the inequality (A.2). We simply observe that $-r^{k+1} + B(z^{k+1} - z^k) = -A(x^{k+1} - x^\star)$; substituting this into (A.2) yields (3.11),

$$p^{k+1} - p^\star \leq -(y^{k+1})^T r^{k+1} + (x^{k+1} - x^\star)^T s^{k+1}.$$

Proof of inequality (A.3)

Since $(x^\star, z^\star, y^\star)$ is a saddle point for L_0, we have

$$L_0(x^\star, z^\star, y^\star) \leq L_0(x^{k+1}, z^{k+1}, y^\star).$$

Using $Ax^\star + Bz^\star = c$, the lefthand side is $p^\star$. With $p^{k+1} = f(x^{k+1}) + g(z^{k+1})$, this can be written as

$$p^\star \leq p^{k+1} + y^{\star T} r^{k+1},$$

which gives (A.3).

Proof of inequality (A.2)

By definition, x^{k+1} minimizes $L_\rho(x, z^k, y^k)$. Since f is closed, proper, and convex it is subdifferentiable, and so is L_ρ. The (necessary and sufficient) optimality condition is

$$0 \in \partial L_\rho(x^{k+1}, z^k, y^k) = \partial f(x^{k+1}) + A^T y^k + \rho A^T (Ax^{k+1} + Bz^k - c).$$

(Here we use the basic fact that the subdifferential of the sum of a subdifferentiable function and a differentiable function with domain $\mathbf{R}^n$ is the sum of the subdifferential and the gradient; see, *e.g.*, [140, §23].)

Since $y^{k+1} = y^k + \rho r^{k+1}$, we can plug in $y^k = y^{k+1} - \rho r^{k+1}$ and rearrange to obtain

$$0 \in \partial f(x^{k+1}) + A^T (y^{k+1} - \rho B(z^{k+1} - z^k)).$$

This implies that x^{k+1} minimizes

$$f(x) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax.$$

A similar argument shows that z^{k+1} minimizes $g(z) + y^{(k+1)T} Bz$. It follows that

$$\begin{aligned} f(x^{k+1}) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax^{k+1} \\ \leq f(x^\star) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax^\star \end{aligned}$$

and that

$$g(z^{k+1}) + y^{(k+1)T} Bz^{k+1} \leq g(z^\star) + y^{(k+1)T} Bz^\star.$$

Adding the two inequalities above, using $Ax^\star + Bz^\star = c$, and rearranging, we obtain (A.2).

Proof of inequality (A.1)

Adding (A.2) and (A.3), regrouping terms, and multiplying through by 2 gives

$$\begin{aligned} 2(y^{k+1}-y^\star)^T r^{k+1} - 2\rho(B(z^{k+1}-z^k))^T r^{k+1} \\ + 2\rho(B(z^{k+1}-z^k))^T(B(z^{k+1}-z^\star)) \leq 0. \end{aligned} \tag{A.4}$$

The result (A.1) will follow from this inequality after some manipulation and rewriting.

We begin by rewriting the first term. Substituting $y^{k+1} = y^k + \rho r^{k+1}$ gives

$$2(y^k - y^\star)^T r^{k+1} + \rho\|r^{k+1}\|_2^2 + \rho\|r^{k+1}\|_2^2,$$

and substituting $r^{k+1} = (1/\rho)(y^{k+1} - y^k)$ in the first two terms gives

$$(2/\rho)(y^k - y^\star)^T(y^{k+1} - y^k) + (1/\rho)\|y^{k+1} - y^k\|_2^2 + \rho\|r^{k+1}\|_2^2.$$

Since $y^{k+1} - y^k = (y^{k+1} - y^\star) - (y^k - y^\star)$, this can be written as

$$(1/\rho)\left(\|y^{k+1} - y^\star\|_2^2 - \|y^k - y^\star\|_2^2\right) + \rho\|r^{k+1}\|_2^2. \tag{A.5}$$

We now rewrite the remaining terms, *i.e.*,

$$\rho\|r^{k+1}\|_2^2 - 2\rho(B(z^{k+1}-z^k))^T r^{k+1} + 2\rho(B(z^{k+1}-z^k))^T(B(z^{k+1}-z^\star)),$$

where $\rho\|r^{k+1}\|_2^2$ is taken from (A.5). Substituting

$$z^{k+1} - z^\star = (z^{k+1} - z^k) + (z^k - z^\star)$$

in the last term gives

$$\begin{aligned} \rho\|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho\|B(z^{k+1} - z^k)\|_2^2 \\ +2\rho(B(z^{k+1} - z^k))^T(B(z^k - z^\star)), \end{aligned}$$

and substituting

$$z^{k+1} - z^k = (z^{k+1} - z^\star) - (z^k - z^\star)$$

in the last two terms, we get

$$\rho\|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho\left(\|B(z^{k+1} - z^\star)\|_2^2 - \|B(z^k - z^\star)\|_2^2\right).$$

With the previous step, this implies that (A.4) can be written as

$$V^k - V^{k+1} \geq \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2. \tag{A.6}$$

To show (A.1), it now suffices to show that the middle term $-2\rho r^{(k+1)T}(B(z^{k+1} - z^k))$ of the expanded right hand side of (A.6) is positive. To see this, recall that z^{k+1} minimizes $g(z) + y^{(k+1)T}Bz$ and z^k minimizes $g(z) + y^{kT}Bz$, so we can add

$$g(z^{k+1}) + y^{(k+1)T}Bz^{k+1} \leq g(z^k) + y^{(k+1)T}Bz^k$$

and

$$g(z^k) + y^{kT}Bz^k \leq g(z^{k+1}) + y^{kT}Bz^{k+1}$$

to get that

$$(y^{k+1} - y^k)^T(B(z^{k+1} - z^k)) \leq 0.$$

Substituting $y^{k+1} - y^k = \rho r^{k+1}$ gives the result, since $\rho > 0$.

References

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.

[2] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems," *IEEE Transactions on Image Processing*, vol. 20, pp. 681–695, 2011.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorenson, *LAPACK: A portable linear algebra library for high-performance computers*. IEEE Computing Society Press, 1990.

[4] K. J. Arrow and G. Debreu, "Existence of an equilibrium for a competitive economy," *Econometrica: Journal of the Econometric Society*, vol. 22, no. 3, pp. 265–290, 1954.

[5] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming*. Stanford University Press: Stanford, 1958.

[6] K. J. Arrow and R. M. Solow, "Gradient methods for constrained maxima, with weakened assumptions," in *Studies in Linear and Nonlinear Programming*, (K. J. Arrow, L. Hurwicz, and H. Uzawa, eds.), Stanford University Press: Stanford, 1958.

[7] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[8] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[9] H. H. Bauschke and J. M. Borwein, "Dykstra's alternating projection algorithm for two sets," *Journal of Approximation Theory*, vol. 79, no. 3, pp. 418–443, 1994.

[10] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Review*, vol. 38, no. 3, pp. 367–426, 1996.

[11] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[12] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," Available at `http://www.acm.caltech.edu/~emmanuel/papers/NESTA.pdf`, 2009.

[13] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, 1962.

[14] A. Bensoussan, J.-L. Lions, and R. Temam, "Sur les méthodes de décomposition, de décentralisation et de coordination et applications," *Methodes Mathematiques de l'Informatique*, pp. 133–257, 1976.

[15] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[16] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, second ed., 1999.

[17] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[18] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Alternating Direction Algorithms for Constrained Sparse Regression: Application to Hyperspectral Unmixing," *arXiv:1002.4527*, 2010.

[19] J. Borwein and A. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Canadian Mathematical Society, 2000.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[21] L. M. Bregman, "Finding the common point of convex sets by the method of successive projections," *Proceedings of the USSR Academy of Sciences*, vol. 162, no. 3, pp. 487–490, 1965.

[22] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, 1967.

[23] H. Brézis, *Opérateurs Maximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hilbert*. North-Holland: Amsterdam, 1973.

[24] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[25] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "HaLoop: Efficient Iterative Data Processing on Large Clusters," *Proceedings of the 36th International Conference on Very Large Databases*, 2010.

[26] R. H. Byrd, P. Lu, and J. Nocedal, "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[27] E. J. Candès and Y. Plan, "Near-ideal model selection by ℓ_1 minimization," *Annals of Statistics*, vol. 37, no. 5A, pp. 2145–2177, 2009.

[28] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, p. 489, 2006.

[29] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[30] Y. Censor and S. A. Zenios, "Proximal minimization algorithm with D-functions," *Journal of Optimization Theory and Applications*, vol. 73, no. 3, pp. 451–464, 1992.

[31] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1997.

[32] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "BigTable: A distributed storage system for structured data," *ACM Transactions on Computer Systems*, vol. 26, no. 2, pp. 1–26, 2008.

[33] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Mathematical Programming*, vol. 64, pp. 81–101, 1994.

[34] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, pp. 129–159, 2001.

[35] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate," *ACM Transactions on Mathematical Software*, vol. 35, no. 3, p. 22, 2008.

[36] W. Cheney and A. A. Goldstein, "Proximity maps for convex sets," *Proceedings of the American Mathematical Society*, vol. 10, no. 3, pp. 448–450, 1959.

[37] C. T. Chu, S. K. Kim, Y. A. Lin, Y. Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, "MapReduce for machine learning on multicore," in *Advances in Neural Information Processing Systems*, 2007.

[38] J. F. Claerbout and F. Muir, "Robust modeling with erratic data," *Geophysics*, vol. 38, p. 826, 1973.

[39] P. L. Combettes, "The convex feasibility problem in image recovery," *Advances in Imaging and Electron Physics*, vol. 95, pp. 155–270, 1996.

[40] P. L. Combettes and J. C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 564–574, 2007.

[41] P. L. Combettes and J. C. Pesquet, "Proximal Splitting Methods in Signal Processing," *arXiv:0912.3522*, 2009.

[42] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2006.

[43] G. B. Dantzig, *Linear Programming and Extensions*. RAND Corporation, 1963.

[44] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, pp. 101–111, 1960.

[45] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, pp. 1413–1457, 2004.

[46] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[47] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM: Philadelphia, PA, 1997.

[48] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, no. 1, pp. 157–175, 1972.

[49] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, pp. 613–627, 1995.

[50] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[51] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, p. 18914, 2009.

[52] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," Tech. Rep., Stanford University, 2006.

[53] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Transactions of the American Mathematical Society*, vol. 82, pp. 421–439, 1956.

[54] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Distributed Dual Averaging in Networks," in *Advances in Neural Information Processing Systems*, 2010.

[55] J. C. Duchi, S. Gould, and D. Koller, "Projected subgradient methods for learning sparse Gaussians," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2008.

[56] R. L. Dykstra, "An algorithm for restricted least squares regression," *Journal of the American Statistical Association*, vol. 78, pp. 837–842, 1983.

[57] J. Eckstein, *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, MIT, 1989.

[58] J. Eckstein, "Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming," *Mathematics of Operations Research*, pp. 202–226, 1993.

[59] J. Eckstein, "Parallel alternating direction multiplier decomposition of convex programs," *Journal of Optimization Theory and Applications*, vol. 80, no. 1, pp. 39–62, 1994.

[60] J. Eckstein, "Some saddle-function splitting methods for convex programming," *Optimization Methods and Software*, vol. 4, no. 1, pp. 75–83, 1994.

[61] J. Eckstein, "A practical general approximation criterion for methods of multipliers based on Bregman distances," *Mathematical Programming*, vol. 96, no. 1, pp. 61–86, 2003.

[62] J. Eckstein and D. P. Bertsekas, "An alternating direction method for linear programming," Tech. Rep., MIT, 1990.

[63] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[64] J. Eckstein and M. C. Ferris, "Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control," *INFORMS Journal on Computing*, vol. 10, pp. 218–235, 1998.

[65] J. Eckstein and M. Fukushima, "Some reformulations and applications of the alternating direction method of multipliers," *Large Scale Optimization: State of the Art*, pp. 119–138, 1993.

[66] J. Eckstein and B. F. Svaiter, "A family of projective splitting methods for the sum of two maximal monotone operators," *Mathematical Programming*, vol. 111, no. 1-2, p. 173, 2008.

[67] J. Eckstein and B. F. Svaiter, "General projective splitting methods for sums of maximal monotone operators," *SIAM Journal on Control and Optimization*, vol. 48, pp. 787–811, 2009.

[68] E. Esser, "Applications of Lagrangian-based alternating direction methods and connections to split Bregman," *CAM report*, vol. 9, p. 31, 2009.

[69] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, 1963.

[70] M. J. Fadili and J. L. Starck, "Monotone operator splitting for optimization problems in sparse recovery," *IEEE ICIP*, 2009.

[71] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Society for Industrial and Applied Mathematics, 1990. First published in 1968 by Research Analysis Corporation.

[72] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian Images Using Alternating Direction Optimization," *IEEE Transactions on Image Processing*, vol. 19, pp. 3133–3145, 2010.

[73] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[74] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[75] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*. North-Holland: Amsterdam, 1983.

[76] M. Fortin and R. Glowinski, "On decomposition-coordination methods using an augmented Lagrangian," in *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*, (M. Fortin and R. Glowinski, eds.), North-Holland: Amsterdam, 1983.

[77] M. Forum, *MPI: A Message-Passing Interface Standard, version 2.2*. High-Performance Computing Center: Stuttgart, 2009.

[78] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*, pp. 23–37, Springer, 1995.

[79] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, p. 432, 2008.

[80] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, vol. 1, pp. 93–111, 1992.

[81] D. Gabay, "Applications of the method of multipliers to variational inequalities," in *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*, (M. Fortin and R. Glowinski, eds.), North-Holland: Amsterdam, 1983.

[82] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[83] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi, *GNU Scientific Library Reference Manual*. Network Theory Ltd., third ed., 2002.

[84] A. M. Geoffrion, "Generalized Benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.

[85] S. Ghemawat, H. Gobioff, and S. T. Leung, "The Google file system," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 29–43, 2003.

[86] R. Glowinski and A. Marrocco, "Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de Dirichlet non lineares," *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, vol. 9, pp. 41–76, 1975.

[87] R. Glowinski and P. L. Tallec, "Augmented Lagrangian methods for the solution of variational problems," Tech. Rep. 2965, University of Wisconsin-Madison, 1987.

[88] T. Goldstein and S. Osher, "The split Bregman method for ℓ_1 regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[89] E. G. Gol'shtein and N. V. Tret'yakov, "Modified Lagrangians in convex programming and their generalizations," *Point-to-Set Maps and Mathematical Programming*, pp. 86–97, 1979.

[90] G. H. Golub and C. F. van Loan, *Matrix Computations*. Johns Hopkins University Press, third ed., 1996.

[91] D. Gregor and A. Lumsdaine, "The Parallel BGL: A generic library for distributed graph computations," *Parallel Object-Oriented Scientific Computing*, 2005.

[92] A. Halevy, P. Norvig, and F. Pereira, "The Unreasonable Effectiveness of Data," *IEEE Intelligent Systems*, vol. 24, no. 2, 2009.

[93] K. B. Hall, S. Gilpin, and G. Mann, "MapReduce/BigTable for distributed optimization," in *Neural Information Processing Systems: Workshop on Learning on Cores, Clusters, and Clouds*, 2010.

[94] T. Hastie and R. Tibshirani, *Generalized Additive Models*. Chapman & Hall, 1990.

[95] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, second ed., 2009.

[96] B. S. He, H. Yang, and S. L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, no. 2, pp. 337–356, 2000.

[97] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 302–320, 1969.

[98] M. R. Hestenes, "Multiplier and gradient methods," in *Computing Methods in Optimization Problems*, (L. A. Zadeh, L. W. Neustadt, and A. V. Balakrishnan, eds.), Academic Press, 1969.

[99] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. Springer, 2001.

[100] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, pp. 73–101, 1964.

[101] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 Trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[102] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[103] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 1, no. 8, pp. 1519–1555, 2007.

[104] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[105] S. A. Kontogiorgis, *Alternating directions methods for the parallel solution of large-scale block-structured optimization problems*. PhD thesis, University of Wisconsin-Madison, 1994.

[106] S. A. Kontogiorgis and R. R. Meyer, "A variable-penalty alternating directions method for convex optimization," *Mathematical Programming*, vol. 83, pp. 29–53, 1998.

[107] L. S. Lasdon, *Optimization Theory for Large Systems*. MacMillan, 1970.

[108] J. Lawrence and J. E. Spingarn, "On fixed points of non-expansive piecewise isometric mappings," *Proceedings of the London Mathematical Society*, vol. 3, no. 3, p. 605, 1987.

[109] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, "Basic linear algebra subprograms for Fortran usage," *ACM Transactions on Mathematical Software*, vol. 5, no. 3, pp. 308–323, 1979.

[110] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, 2001.

[111] J. Lin and M. Schatz, "Design Patterns for Efficient Graph Algorithms in MapReduce," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pp. 78–85, 2010.

[112] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, pp. 964–979, 1979.

[113] D. C. Liu and J. Nocedal, "On the Limited Memory Method for Large Scale Optimization," *Mathematical Programming B*, vol. 45, no. 3, pp. 503–528, 1989.

[114] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, "GraphLab: A New Parallel Framework for Machine Learning," in *Conference on Uncertainty in Artificial Intelligence*, 2010.

[115] Z. Lu, "Smooth optimization approach for sparse covariance selection," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1807–1827, 2009.

[116] Z. Lu, T. K. Pong, and Y. Zhang, "An Alternating Direction Method for Finding Dantzig Selectors," `arXiv:1011.4604`, 2010.

[117] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Addison-Wesley: Reading, MA, 1973.

[118] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network flow algorithms for structured sparsity," *Advances in Neural Information Processing Systems*, vol. 24, 2010.

[119] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing," in *Proceedings of the 2010 International Conference on Management of Data*, pp. 135–146, 2010.

[120] A. F. T. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing, "An Augmented Lagrangian Approach to Constrained MAP Inference," in *International Conference on Machine Learning*, 2011.

[121] G. Mateos, J.-A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5262–5276, Oct. 2010.

[122] P. J. McCullagh and J. A. Nelder, *Generalized Linear Models*. Chapman & Hall, 1991.

[123] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the lasso," *Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[124] A. Miele, E. E. Cragg, R. R. Iver, and A. V. Levy, "Use of the augmented penalty function in mathematical programming problems, part 1," *Journal of Optimization Theory and Applications*, vol. 8, pp. 115–130, 1971.

[125] A. Miele, E. E. Cragg, and A. V. Levy, "Use of the augmented penalty function in mathematical programming problems, part 2," *Journal of Optimization Theory and Applications*, vol. 8, pp. 131–153, 1971.

[126] A. Miele, P. E. Mosely, A. V. Levy, and G. M. Coggins, "On the method of multipliers for mathematical programming problems," *Journal of Optimization Theory and Applications*, vol. 10, pp. 1–33, 1972.

[127] J.-J. Moreau, "Fonctions convexes duales et points proximaux dans un espace Hilbertien," *Reports of the Paris Academy of Sciences, Series A*, vol. 255, pp. 2897–2899, 1962.

[128] D. Mosk-Aoyama, T. Roughgarden, and D. Shah, "Fully distributed algorithms for convex optimization problems," Available at `http://theory.stanford.edu/~tim/papers/distrcvxopt.pdf`, 2007.

[129] I. Necoara and J. A. K. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[130] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[131] A. Nedić and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications*, (D. P. Palomar and Y. C. Eldar, eds.), Cambridge University Press, 2010.

[132] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[133] Y. Nesterov, "Gradient methods for minimizing composite objective function," *CORE Discussion Paper, Catholic University of Louvain*, vol. 76, p. 2007, 2007.

[134] M. Ng, P. Weiss, and X. Yuang, "Solving Constrained Total-Variation Image Restoration and Reconstruction Problems via Alternating Direction Methods," *ICM Research Report*, Available at `http://www.optimization-online.org/DB_FILE/2009/10/2434.pdf`, 2009.

[135] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, 1999.

[136] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of ARX-models using sum-of-norms regularization," *Automatica*, vol. 46, pp. 1107–1111, 2010.

[137] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, pp. 28–41, 1955.

[138] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, (R. Fletcher, ed.), Academic Press, 1969.

[139] A. Ribeiro, I. Schizas, S. Roumeliotis, and G. Giannakis, "Kalman filtering in wireless sensor networks — Incorporating communication cost in state estimation problems," *IEEE Control Systems Magazine*, vol. 30, pp. 66–86, Apr. 2010.

[140] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[141] R. T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, pp. 97–116, 1976.

[142] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, p. 877, 1976.

[143] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119–147, 1991.

[144] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer-Verlag, 1998.

[145] L. Rudin, S. J. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[146] A. Ruszczyński, "An augmented Lagrangian decomposition method for block diagonal linear programming problems," *Operations Research Letters*, vol. 8, no. 5, pp. 287–294, 1989.

[147] A. Ruszczyński, "On convergence of an augmented Lagrangian decomposition method for sparse convex optimization," *Mathematics of Operations Research*, vol. 20, no. 3, pp. 634–656, 1995.

[148] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," in *Advances in Neural Information Processing Systems*, 2010.

[149] I. D. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links — part II: Distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 1650–1666, Apr. 2008.

[150] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links — part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 350–364, Jan. 2008.

[151] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[152] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, 1985.

[153] J. E. Spingarn, "Applications of the method of partial inverses to convex programming: decomposition," *Mathematical Programming*, vol. 32, pp. 199–223, 1985.

[154] G. Steidl and T. Teuber, "Removing multiplicative noise by Douglas-Rachford splitting methods," *Journal of Mathematical Imaging and Vision*, vol. 36, no. 2, pp. 168–184, 2010.

[155] C. H. Teo, S. V. N. Vishwanathan, A. J. Smola, and Q. V. Le, "Bundle methods for regularized risk minimization," *Journal of Machine Learning Research*, vol. 11, pp. 311–365, 2010.

[156] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1996.

[157] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities.," *SIAM Journal on Control and Optimization*, vol. 29, pp. 119–138, 1991.

[158] P. Tseng, "Alternating projection-proximal methods for convex programming and variational inequalities," *SIAM Journal on Optimization*, vol. 7, pp. 951–965, 1997.

[159] P. Tseng, "A modified forward-backward splitting method for maximal monotone mappings," *SIAM Journal on Control and Optimization*, vol. 38, p. 431, 2000.

[160] J. N. Tsitsiklis, *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[161] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[162] H. Uzawa, "Market mechanisms and mathematical programming," *Econometrica: Journal of the Econometric Society*, vol. 28, no. 4, pp. 872–881, 1960.

[163] H. Uzawa, "Walras' tâtonnement in the theory of exchange," *The Review of Economic Studies*, vol. 27, no. 3, pp. 182–194, 1960.

[164] L. G. Valiant, "A bridging model for parallel computation," *Communications of the ACM*, vol. 33, no. 8, p. 111, 1990.

[165] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

[166] J. von Neumann, *Functional Operators, Volume 2: The Geometry of Orthogonal Spaces*. Princeton University Press: Annals of Mathematics Studies, 1950. Reprint of 1933 lecture notes.

[167] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[168] L. Walras, *Éléments d'économie politique pure, ou, Théorie de la richesse sociale*. F. Rouge, 1896.

[169] S. L. Wang and L. Z. Liao, "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *Journal of Optimization Theory and Applications*, vol. 109, no. 2, pp. 415–429, 2001.

[170] T. White, *Hadoop: The Definitive Guide*. O'Reilly Press, second ed., 2010.

[171] J. M. Wooldridge, *Introductory Econometrics: A Modern Approach*. South Western College Publications, fourth ed., 2009.

[172] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[173] A. Y. Yang, A. Ganesh, Z. Zhou, S. S. Sastry, and Y. Ma, "A Review of Fast ℓ_1-Minimization Algorithms for Robust Face Recognition," *arXiv:1007.3753*, 2010.

[174] J. Yang and X. Yuan, "An inexact alternating direction method for trace norm regularized least squares problem," Available at `http://www.optimization-online.org`, 2010.

[175] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *Preprint*, 2009.

[176] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.

[177] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[178] X. M. Yuan, "Alternating direction methods for sparse covariance selection," *Preprint*, Available at `http://www.optimization-online.org/DB_FILE/2009/09/2390.pdf`, 2009.

[179] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, 2010.

[180] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Annals of Statistics*, vol. 32, no. 1, pp. 56–85, 2004.

[181] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[182] H. Zhu, A. Cano, and G. B. Giannakis, "Distributed consensus-based demodulation: algorithms and error analysis," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 2044–2054, 2010.

[183] H. Zhu, G. B. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3970–3983, 2009.