Optimization for Machine Learning

Editors:

Suvrit Sra suvrit@gmail.com
Max Planck Insitute for Biological Cybernetics
72076 Tübingen, Germany

Sebastian Nowozin nowozin@gmail.com
Microsoft Research
Cambridge, CB3 0FB, United Kingdom

Stephen J. Wright swright@cs.uwisc.edu
University of Wisconsin
Madison, WI 53706

This is a draft containing only `sra_chapter.tex` and an abbreviated front matter. Please check that the formatting and small changes have been performed correctly. Please verify the affiliation. Please use this version for sending us future modifications.

The MIT Press
Cambridge, Massachusetts
London, England

Contents

1 Augmented Lagrangian Methods for Learning, Selecting, and Combining Features

Ryota Tomioka tomioka@mist.i.u-tokyo.ac.jp
The University of Tokyo
Tokyo, Japan

Taiji Suzuki s-taiji@stat.t.u-tokyo.ac.jp
The University of Tokyo
Tokyo, Japan

Masashi Sugiyama sugi@cs.titech.ac.jp
Tokyo Institute of Technology
Tokyo, Japan

We investigate the family of Augmented Lagrangian (AL) methods for minimizing the sum of two convex functions. In the context of machine learning, minimization of such a composite *objective function is useful in enforcing various structures, for instance sparsity, on the solution in a learning task. We introduce a particularly efficient instance of an augmented Lagrangian method called the Dual Augmented-Lagrangian (DAL) algorithm and discuss its connection to proximal minimization and operator splitting algorithms in the primal. Furthermore, we demonstrate that the DAL algorithm for the trace norm regularization can be used to learn features from multiple data sources and combine them in an optimal way in a convex optimization problem.*

1.1 Introduction

Sparse estimation has recently been attracting attention from both theoretical side (Candes et al., 2006; Bach, 2008; Ng, 2004) and practical side, for example, magnetic resonance imaging (Weaver et al., 1991; Lustig et al., 2007), natural language processing (Gao et al., 2007), and bioinformatics (Shevade and Keerthi, 2003).

Sparse estimation is commonly formulated in two ways: the regularized estimation (or MAP estimation) framework (Tibshirani, 1996), and the empirical Bayesian estimation (also known as the automatic relevance determination) (Neal, 1996; Tipping, 2001). Both approaches are based on optimizing some objective functions, though the former is usually formulated as a convex optimization and the later is usually nonconvex.

Recently, a connection between the two formulations has been discussed in Wipf and Nagarajan (2008), which showed that in some special cases the (nonconvex) empirical Bayesian estimation can be carried out by iteratively solving reweighted (convex) regularized estimation problems. Therefore, in this chapter we will focus on the convex approach.

Regularized estimation

A regularization-based sparse-estimation problem can be formulated as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \underbrace{L(\boldsymbol{x}) + R(\boldsymbol{x})}_{=:f(\boldsymbol{x})}, \tag{1.1}$$

where $L : \mathbb{R}^n \to \mathbb{R}$ is called the loss term, which we assume to be convex and differentiable, $R : \mathbb{R}^n \to \mathbb{R}$ is called the regularizer, which is assumed to be convex but may be non-differentiable, and for convenience we denote by f the sum of the two. In addition, we assume that $f(\boldsymbol{x}) \to \infty$ as $\|\boldsymbol{x}\| \to \infty$.

Finding a zero of the sum of two operators

Problem (1.1) is closely related to solving an operator equation

$$(A + B)(\boldsymbol{x}) \ni 0, \tag{1.2}$$

where A and B are nonlinear maximal monotone operators. In fact, if A and B are the subdifferentials of L and R, respectively, the two problems (1.1) and (1.2) are equivalent. Algorithms to solve the operator equation (1.2) are extensively studied and are called *operator splitting* methods; see Lions and Mercier (1979); Eckstein and Bertsekas (1992). We will discuss their connections to minimization algorithms for (1.1) in Sections 1.2.2 and 1.5.

Simple sparse estimation problem

We will make distinction between a *simple* sparse estimation problem, and a *structured* sparse estimation problem. A simple sparse estimation problem is written as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad L(\boldsymbol{x}) + \phi_\lambda(\boldsymbol{x}), \tag{1.3}$$

where ϕ_λ is a closed proper convex function[1], and is "simple" in the sense of separability and sparsity, which we define in Section 1.2.1. Examples of a simple sparse estimation problem include the lasso (Tibshirani, 1996) (also known as the basis-pursuit denoising (Chen et al., 1998)), the group lasso (Yuan and Lin, 2006) and the trace norm regularization (Fazel et al., 2001; Srebro et al., 2005; Tomioka and Aihara, 2007; Yuan et al., 2007).

Structured sparse estimation problem

A *structured* sparse estimation problem is written as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad L(\boldsymbol{x}) + \phi_\lambda(\boldsymbol{B}\boldsymbol{x}), \tag{1.4}$$

where $\boldsymbol{B} \in \mathbb{R}^{l\times n}$ is a matrix and ϕ_λ is a simple sparse regularizer as in the simple sparse estimation problem (1.3). Examples of a structured sparse estimation problem include the total-variation denoising (Rudin et al., 1992), wavelet shrinkage (Weaver et al., 1991; Donoho, 1995), fused lasso (Tibshirani et al., 2005), and the structured sparsity inducing norms (Jenatton et al., 2009).

In this chapter, we present an augmented Lagrangian (AL) method (Hestenes, 1969; Powell, 1969) for the dual of the simple sparse estimation problem. We show that the proposed *dual* augmented Lagrangian (DAL) is equivalent to the proximal minimization algorithm in the primal, converges super-linearly[2], and each step is computationally efficient because DAL can exploit the *sparsity in the intermediate solution.* Note that there has been a series of studies that derived AL approaches using Bregman divergence; see Yin et al. (2008); Cai et al. (2008); Setzer (2010).

Although our focus will be mostly on the simple sparse estimation problem (1.3), the methods we discuss are also relevant for the structured sparse estimation problem (1.4). In fact, by taking the Fenchel dual (Rockafellar, 1970, Theorem 31.2), we notice that solving the structured sparse estimation problem (1.4) is equivalent to solving the following minimization problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^l} \quad L^*(\boldsymbol{B}^T\boldsymbol{\beta}) + \phi_\lambda^*(-\boldsymbol{\beta}), \tag{1.5}$$

[1] "Closed" means that the epigraph $\{(\boldsymbol{z}, y) \in \mathbb{R}^{m+1} : y \geq \phi_\lambda(\boldsymbol{z})\}$ is a closed set, and "proper" means that the function is not everywhere $+\infty$; see e.g., Rockafellar (1970). In the sequel, we use the word "convex function" in the meaning of "closed proper convex function".

[2] A sequence $\boldsymbol{x}^t$ $(t = 1, 2, \ldots)$ converges to $\boldsymbol{x}^*$ super-linearly, if $\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq c_t\|\boldsymbol{x}^t - \boldsymbol{x}^*\|$, where $0 \leq c_t < 1$ and $c_t \to 0$ as $t \to \infty$.

where L^* and ϕ_λ^* are the convex conjugate functions of L and ϕ_λ, respectively. The above minimization problem resembles the simple sparse estimation problem (1.3) (the matrix $\boldsymbol{B}^T$ can be considered as part of the loss function). This fact was effectively used by Goldstein and Osher (2009) to develop the split Bregman iteration (SBI) algorithm (see also Setzer (2010)). See Section 1.5 for more detailed discussion.

Organization

This chapter is organized as follows. In the next section, we introduce some simple sparse regularizers and review different types of sparsity they produce through the so called proximity operator. A brief operator theoretic background for the proximity operator is also given. In Section 1.3, we present the proximal minimization algorithm, which is the primal representation of DAL algorithm. The proposed DAL algorithm is introduced in Section 1.4 and we discuss why the dual formulation is particularly suitable for the simple sparse estimation problem and discuss its rate of convergence. We discuss connections between approximate AL methods and two operator splitting algorithms, namely the forward-backward splitting and the Douglas-Rachford splitting in Section 1.5. In Section 1.6, we apply the trace norm regularization to a real brain-computer interface dataset for learning feature extractors and their optimal combination. The computational efficiency of DAL algorithm is also demonstrated. Finally we summarize this chapter in Section 1.7. Some background material on convex analysis is given in Appendix.

1.2 Background

In this section, we define “simple” sparse regularizers through the associated proximity operators. In addition, Section 1.2.2 provides some operator theoretic backgrounds, which we use in later sections, especially in Section 1.5.

1.2.1 Simple sparse regularizers

In this section, we provide three examples of *simple* sparse regularizers, namely, the ℓ_1-regularizer, the group lasso regularizer, and the trace norm regularizer. Other regularizers obtained by applying these three regularizers in a block-wise manner will also be called simple; for example, the ℓ_1-regularizer for the first 10 variables and the group lasso regularizer for the remaining variables. These regularizers share two important properties. First, they are *separable* (in some manner). Second, the so-called proximity operators they define return “sparse” vectors (with respect to their separability).

Proximity operator

We need to first define the proximity operator as below; see also Moreau (1965); Rockafellar (1970); Combettes and Wajs (2005).

Definition 1.1. *The proximity operator corresponding to a convex function $f : \mathbb{R}^n \to \mathbb{R}$ over $\mathbb{R}^n$ is a mapping from $\mathbb{R}^n$ to itself and is defined as*

$$\operatorname{prox}_f(\boldsymbol{z}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \left(f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|^2 \right), \tag{1.6}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Note that the minimizer is unique because the objective is strongly convex. Although the above definition is given in terms of a function f over $\mathbb{R}^n$, the definition extends naturally to a function over a general Hilbert space; see Moreau (1965); Rockafellar (1970).

The proximity operator (1.6) defines a unique decomposition of a vector $\boldsymbol{z}$ as

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y},$$

where $\boldsymbol{x} = \operatorname{prox}_f(\boldsymbol{z})$ and $\boldsymbol{y} = \operatorname{prox}_{f^*}(\boldsymbol{z})$ (f^* is the convex conjugate of f). This is called Moreau's decomposition; see Appendix 1.B. We denote Moreau's decomposition corresponding to the function f as follows:

$$(\boldsymbol{x}, \boldsymbol{y}) = \operatorname{decomp}_f(\boldsymbol{z}). \tag{1.7}$$

Note that the above expression implies $\boldsymbol{y} \in \partial f(\boldsymbol{x})$ because $\boldsymbol{x}$ minimizes the objective (1.6) and $\partial f(\boldsymbol{x}) + \boldsymbol{x} - \boldsymbol{z} \ni 0$, where $\partial f(\boldsymbol{x})$ denotes the subdifferential of f at $\boldsymbol{x}$.

ℓ_1-regularizer

The first example of sparse regularizers is the ℓ_1-regularizer (or the lasso regularizer (Tibshirani, 1996)), which is defined as follows:

$$\phi_\lambda^{\ell_1}(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1 = \lambda\sum_{j=1}^{n} |x_j|, \tag{1.8}$$

where $|\cdot|$ denotes the absolute value. We can also allow each component to have different regularization constant, which can be used to include an unregularized bias term.

The proximity operator corresponding to the ℓ_1-regularizer is known as the soft-threshold operator (Donoho, 1995) and can be defined elementwise as follows:

$$\operatorname{prox}_\lambda^{\ell_1}(\boldsymbol{z}) := \left(\max(|z_j| - \lambda, 0)\frac{z_j}{|z_j|} \right)_{j=1}^{n}, \tag{1.9}$$

where the ratio $z_j/|z_j|$ is defined to be zero if $z_j = 0$. The above expression can easily be derived because the objective (1.6) can be minimized for each component x_j independently for the ℓ_1-regularizer.

Group lasso regularizer

The second example of sparse regularizers is the group-lasso (Yuan and Lin, 2006) regularizer

$$\phi_\lambda^{\mathfrak{G}}(\boldsymbol{x}) = \lambda \sum_{\mathfrak{g} \in \mathfrak{G}} \|\boldsymbol{x}_{\mathfrak{g}}\|, \tag{1.10}$$

where $\mathfrak{G}$ is a non-overlapping partition of $\{1, \ldots, n\}$, $\mathfrak{g} \in \mathfrak{G}$ is an index-set $\mathfrak{g} \subseteq \{1, \ldots, n\}$, and $\boldsymbol{x}_{\mathfrak{g}}$ is a sub-vector of $\boldsymbol{x}$ specified by the indices in $\mathfrak{g}$. For example, the group-lasso regularizer arises when we are estimating a vector field on a grid over a two-dimensional vector space. Shrinking each component of the vectors individually through ℓ_1-regularization can produce vectors either pointing along the x-axis or the y-axis but not necessarily sparse as a vector field. We can group the x- and y-components of the vectors and apply the group lasso regularizer (1.10) to shrink both components of the vectors simultaneously.

The proximity operator corresponding to the group lasso regularizer can be written blockwise as follows:

$$\operatorname{prox}_\lambda^{\mathfrak{G}}(\boldsymbol{z}) := \left(\max(\|\boldsymbol{z}_{\mathfrak{g}}\| - \lambda, 0) \frac{\boldsymbol{z}_{\mathfrak{g}}}{\|\boldsymbol{z}_{\mathfrak{g}}\|} \right)_{\mathfrak{g} \in \mathfrak{G}}, \tag{1.11}$$

where the ratio $\boldsymbol{z}_{\mathfrak{g}} / \|\boldsymbol{z}_{\mathfrak{g}}\|_2$ is defined to be zero if $\|\boldsymbol{z}_{\mathfrak{g}}\|_2$ is zero. The above expression can be derived analogous to the ℓ_1 case, because the objective (1.6) can be minimized for each block and from the Cauchy-Schwarz inequality we have

$$\|\boldsymbol{x}_{\mathfrak{g}} - \boldsymbol{z}_{\mathfrak{g}}\|^2 + \lambda \|\boldsymbol{x}_{\mathfrak{g}}\| \geq (\|\boldsymbol{x}_{\mathfrak{g}}\| - \|\boldsymbol{z}_{\mathfrak{g}}\|)^2 + \lambda \|\boldsymbol{x}_{\mathfrak{g}}\|,$$

where the equality is obtained when $\boldsymbol{x}_{\mathfrak{g}} = c \boldsymbol{z}_{\mathfrak{g}}$; the coefficient c can be obtained by solving the one-dimensional minimization.

Trace norm regularizer

The last example of sparse regularizers is the trace-norm[3] regularizer, which is defined as follows:

$$\phi_\lambda^{\mathrm{mat}}(\boldsymbol{x}) = \lambda \|\boldsymbol{X}\|_* = \lambda \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}), \tag{1.12}$$

where $\boldsymbol{X}$ is a matrix obtained by rearranging the elements of $\boldsymbol{x}$ into a matrix of a prespecified size, $\sigma_j(\boldsymbol{X})$ is the jth largest singular-value of $\boldsymbol{X}$, and r is the minimum of the number of rows and columns of $\boldsymbol{X}$. The proximity

[3] The trace norm is also known as the nuclear norm (Boyd and Vandenberghe, 2004) and the Ky Fan r-norm (Yuan et al., 2007).

operator corresponding to the trace norm regularizer can be written as follows:

$$\mathrm{prox}_{\lambda}^{\mathrm{mat}}(\boldsymbol{z}) := \mathrm{vec}\left(\boldsymbol{U}\max(\boldsymbol{S}-\lambda,0)\boldsymbol{V}^T\right), \tag{1.13}$$

where $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ is the singular-value decomposition of the matrix $\boldsymbol{Z}$ obtained by appropriately rearranging the elements of $\boldsymbol{z}$. The above expression can again be obtained using the separability of ϕ_λ as follows:

$$\begin{aligned}
&\|\boldsymbol{X}-\boldsymbol{Z}\|_F^2 + \lambda\sum_{j=1}^{r}\sigma_j(\boldsymbol{X}) \\
&\quad= \sum_{j=1}^{r}\sigma_j^2(\boldsymbol{X}) - 2\langle\boldsymbol{X},\,\boldsymbol{Z}\rangle + \sum_{j=1}^{r}\sigma_j^2(\boldsymbol{Z}) + \lambda\sum_{j=1}^{r}\sigma_j(\boldsymbol{Z}) \\
&\quad\geq \sum_{j=1}^{r}\sigma_j^2(\boldsymbol{X}) - 2\sum_{j=1}^{r}\sigma_j(\boldsymbol{X})\sigma_j(\boldsymbol{Z}) + \sum_{j=1}^{r}\sigma_j^2(\boldsymbol{Z}) + \lambda\sum_{j=1}^{r}\sigma_j(\boldsymbol{Z}) \\
&\quad= \sum_{j=1}^{r}\Big((\sigma_j(\boldsymbol{X})-\sigma_j(\boldsymbol{Z}))^2 + \lambda\sigma_j(\boldsymbol{Z})\Big),
\end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and the inequality in the second line is due to von Neumann's trace theorem (Horn and Johnson, 1991), for which equality is obtained when the singular vectors of $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the same. Singular-values $\sigma_j(\boldsymbol{X})$ is obtained by the one-dimensional minimization in the last line.

Separability and sparsity

Note again that the above three regularizers are *separable*. The ℓ_1-regularizer (1.8) decomposes into the sum of the absolute values of components of $\boldsymbol{x}$. The group-lasso regularizer (1.10) decomposes into the sum of the Euclidean norms of the groups of variables. Finally the trace norm regularizer (1.12) decomposes into the sum of singular-values. Moreover, the proximity operators they define sparsify vectors with respect to the separability of the regularizers; see Equations (1.9), (1.11), and (1.13).

Note that the "regularizer" in the dual of the structured sparse estimation problem (1.5) is also separable, but the corresponding proximity operator does not sparsify a vector; see Section 1.4.4 for more discussion.

The sparsity produced by the proximity operator (1.6) is a computational advantage of algorithms that iteratively compute the proximity operator; see Figueiredo and Nowak (2003); Daubechies et al. (2004); Combettes and Wajs (2005); Figueiredo et al. (2007); Wright et al. (2009); Beck and Teboulle (2009); Nesterov (2007). Other methods, for example interior point methods (Koh et al., 2007; Kim et al., 2007; Boyd and Vandenberghe, 2004), achieve sparsity only asymptotically.

1.2.2 Monotone operator theory background

The proximity operator has been studied intensively in the context of monotone operator theory. This framework provides alternative view on proximity-operator-based algorithms and forms the foundation of operator splitting algorithms, which we discuss in Section 1.5. In this section, we briefly provide background on monotone operator theory; see Rockafellar (1976a); Lions and Mercier (1979); Eckstein and Bertsekas (1992) for more details.

Monotone operator

A nonlinear set-valued operator $T : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is called *monotone* if $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$,

$$\langle \boldsymbol{y}' - \boldsymbol{y}, \, \boldsymbol{x}' - \boldsymbol{x} \rangle \geq 0, \qquad \text{for all} \quad \boldsymbol{y} \in T(\boldsymbol{x}), \boldsymbol{y}' \in T(\boldsymbol{x}'),$$

where $\langle \boldsymbol{y}, \, \boldsymbol{x} \rangle$ denotes the inner product of two vectors $\boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^n$.

The graph of a set-valued operator T is the set $\{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in T(\boldsymbol{x})\} \subseteq \mathbb{R}^n \times \mathbb{R}^n$. A monotone operator T is called *maximal* if the graph of T is not strictly contained in that of any other monotone operator on $\mathbb{R}^n$. The subdifferential of a convex function over $\mathbb{R}^n$ is an example of maximal monotone operator. A set-valued operator T is called *single-valued* if the set $T(\boldsymbol{x})$ consists of a single vector for every $\boldsymbol{x} \in \mathbb{R}^n$. With a slight abuse of notation we denote $\boldsymbol{y} = T(\boldsymbol{x})$ in this case. The subdifferential of the function f defined over $\mathbb{R}^n$ is single-valued if and only if f is differentiable. The *sum* of two set-valued operators A and B is defined by the graph $\{(\boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z}) : \boldsymbol{y} \in A(\boldsymbol{x}), \boldsymbol{z} \in B(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^n\}$. The *inverse* T^{-1} of a set valued operator T is the operator defined by the graph $\{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in T(\boldsymbol{y}), \, \boldsymbol{y} \in \mathbb{R}^n\}$.

Denoting the subdifferential of the function f by $T_f := \partial f$, we can rewrite the proximity operator (1.6) as

$$\operatorname{prox}_f(\boldsymbol{z}) = (I + T_f)^{-1}(\boldsymbol{z}), \tag{1.14}$$

where I denotes the identity mapping. The above expression can be derived from the optimality condition $T_f(\boldsymbol{x}) + \boldsymbol{x} - \boldsymbol{z} \ni 0$. Note that the above expression is single-valued, because the minimizer defining the proximity operator (1.6) is unique. Moreover, the monotonicity of the operator T_f guarantees that the proximity operator (1.14) is firmly nonexpansive[4]. Furthermore, $\operatorname{prox}_f(\boldsymbol{z}) = \boldsymbol{z}$ if and only if $0 \in T_f(\boldsymbol{z})$, because if $\boldsymbol{z}' = \operatorname{prox}_f(\boldsymbol{z})$, $\boldsymbol{z} - \boldsymbol{z}' \in T_f(\boldsymbol{z}')$ and $0 \leq \langle \boldsymbol{z}' - \boldsymbol{z}, \, \boldsymbol{z} - \boldsymbol{z}' - \boldsymbol{y} \rangle$ for all $\boldsymbol{y} \in T_f(\boldsymbol{z})$.

Proximity operator is firmly nonexpansive

[4] An operator T is called *firmly nonexpansive* if $\|\boldsymbol{y}' - \boldsymbol{y}\|^2 \leq \langle \boldsymbol{x}' - \boldsymbol{x}, \, \boldsymbol{y}' - \boldsymbol{y} \rangle$ holds for all $\boldsymbol{y} \in T(\boldsymbol{x})$, $\boldsymbol{y}' \in T(\boldsymbol{x}')$, $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$. This is clearly stronger than the ordinary nonexpansiveness defined by $\|\boldsymbol{y}' - \boldsymbol{y}\| \leq \|\boldsymbol{x}' - \boldsymbol{x}\|$.

1.3 Proximal minimization algorithm

The proximal minimization algorithm (or the proximal point algorithm) iteratively applies the proximity operator (1.6) to obtain the minimizer of some convex function f. Although it is probably never used in its original form in practice, it functions as a foundation for the analysis of both AL algorithms and operator splitting algorithms.

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function that we wish to minimize. Without loss of generality, we focus on unconstrained minimization of f; minimizing a function $f_0(\boldsymbol{x})$ in a convex set C is equivalent to minimizing $f(\boldsymbol{x}) := f_0(\boldsymbol{x}) + \delta_C(\boldsymbol{x})$ where $\delta_C(\boldsymbol{x})$ is the indicator function of C.

Proximal minimization algorithm

A *proximal minimization algorithm* for minimizing f starts from some initial solution $\boldsymbol{x}^0$ and iteratively solves the minimization problem

$$\boldsymbol{x}^{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \left(f(\boldsymbol{x}) + \frac{1}{2\eta_t}\|\boldsymbol{x} - \boldsymbol{x}^t\|^2 \right). \tag{1.15}$$

The second term in the iteration (1.15) keeps the next iterate $\boldsymbol{x}^{t+1}$ in the proximity of the current iterate $\boldsymbol{x}^t$; the parameter η_t controls the strength of the proximity term. From the above iteration, one can easily see that

$$f(\boldsymbol{x}^{t+1}) \le f(\boldsymbol{x}^t) - \frac{1}{2\eta_t}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|^2.$$

Thus, the objective value $f(\boldsymbol{x}^t)$ decreases monotonically as long as $\boldsymbol{x}^{t+1} \neq \boldsymbol{x}^t$.

The iteration (1.15) can also be expressed in terms of the proximity operator (1.6) as follows:

$$\boldsymbol{x}^{t+1} = \mathrm{prox}_{\eta_t f}(\boldsymbol{x}^t) = (I + \eta_t T_f)^{-1}(\boldsymbol{x}^t), \tag{1.16}$$

which is called the *proximal point algorithm* (Rockafellar, 1976a). Since each step is an application of the proximity operator (1.16), it is a firmly nonexpansive mapping for any choice of η_t [actually any iterative algorithm that uses a firmly nonexpansive mapping can be considered as a proximal point algorithm (Eckstein and Bertsekas, 1992).] Moreover, $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t$ if and only if $0 \in T_f(\boldsymbol{x}^t)$; i.e., $\boldsymbol{x}^t$ is a minimizer of f. The connection between minimizing a convex function and finding a zero of a maximal monotone operator can be summarized as in Table 1.1.

The iteration (1.15) can also be considered as an *implicit gradient step* because

$$\boldsymbol{x}^{t+1} - \boldsymbol{x}^t \in -\eta_t \partial f(\boldsymbol{x}^{t+1}). \tag{1.17}$$

Table 1.1: Comparison of the proximal minimization algorithm for convex optimization and the proximal point algorithm for solving operator equations.

	Convex optimization	Operator equation
Objective	minimize $f(\boldsymbol{x})$	find $0 \in T_f(\boldsymbol{x})$
Algorithm	Proximal minimization algorithm $\boldsymbol{x}^{t+1} = \mathrm{prox}_{\eta_t f}(\boldsymbol{x}^t)$	Proximal point algorithm $\boldsymbol{x}^{t+1} = (I + \eta_t T_f)^{-1}(\boldsymbol{x}^t)$

Note that the subdifferential in the right-hand side is evaluated at the new point $\boldsymbol{x}^{t+1}$.

Convergence

Rockafellar (1976a) has shown under mild assumptions, which also allow errors in the minimization (1.15), that the sequence $\boldsymbol{x}^0, \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$ converges[5] to a point $\boldsymbol{x}^\infty$ that satisfies $0 \in T_f(\boldsymbol{x}^\infty)$. Rockafellar (1976a) has also shown that the convergence of the proximal minimization algorithm is super-linear under the assumption that T_f^{-1} is locally Lipschitz around the origin.

The following theorem states the super-linear convergence of the proximal minimization algorithm in a non-asymptotic sense.

Theorem 1.2. *Let* $\boldsymbol{x}^0, \boldsymbol{x}^1, \boldsymbol{x}^2 \ldots$ *be the sequence generated by the* exact *proximal minimization algorithm* (1.15) *and let* $\boldsymbol{x}^*$ *be a minimizer of the objective function* f*. Assume that there is a positive constant* σ *and a scalar* α *(*$1 \leq \alpha \leq 2$*) such that*

$$\textbf{(A1)} \qquad f(\boldsymbol{x}^{t+1}) - f(\boldsymbol{x}^*) \geq \sigma \|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^\alpha \qquad (t = 0, 1, 2, \ldots).$$

Then the following inequality is true:

$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^{\frac{1+(\alpha-1)\sigma\eta_t}{1+\sigma\eta_t}} \leq \frac{1}{1+\sigma\eta_t}\|\boldsymbol{x}^t - \boldsymbol{x}^*\|.$$

That is, $\boldsymbol{x}^t$ *converges to* $\boldsymbol{x}^*$ *super-linearly if* $\alpha < 2$ *or* $\alpha = 2$ *and* η_t *is increasing, in a global and non-asymptotic sense.*

Proof. See Tomioka et al. (2010a). □

Assumption **(A1)** is implied by assuming the strong convexity of f. However, it is weaker because we only require **(A1)** on the points generated by the algorithm. For example, the ℓ_1-regularizer (1.8) is not strongly convex but it can be lower-bounded as in assumption **(A1)** inside any bounded set centered at the origin. In fact, the assumption on the Lipschitz continuity of ∂f^{-1} around the origin used in Rockafellar (1976b) implies assumption **(A1)**

[5] The original statement was "converges in the weak topology", which is equivalent to strong convergence in a finite dimensional vector space.

due to the nonexpansiveness of the proximity operator (1.16); see Tomioka et al. (2010a) for a detailed discussion.

So far we have ignored the cost of the minimization (1.15). The convergence rate in the above theorem becomes faster as the proximity parameter η_t increases. However, typically the cost of the minimization (1.15) increases as η_t increases. In the next section, we focus on how we can carry out the update step (1.15) efficiently.

1.4 Dual Augmented Lagrangian (DAL) algorithm

In this section, we introduce Dual Augmented Lagrangian (DAL) (Tomioka and Sugiyama, 2009; Tomioka et al., 2010a) and show that it is equivalent to the proximal minimization algorithm we discussed in the previous section. For the simple sparse estimation problem (1.3) each step in DAL is computationally efficient. Thus it is practical and can be analyzed through the proximal minimization framework.

1.4.1 DAL as augmented Lagrangian applied to the dual problem

DAL is an application of the Augmented Lagrangian (AL) algorithm (Hestenes, 1969; Powell, 1969) to the dual of the simple sparse estimation problem

$$\text{(P)} \qquad \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f_\ell(\boldsymbol{A}\boldsymbol{x}) + \phi_\lambda(\boldsymbol{x}), \tag{1.18}$$

Separation of the loss function from the data matrix

where $f_\ell : \mathbb{R}^m \to \mathbb{R}$ is a loss function, which we assume to be a smooth convex function; $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a design matrix. Note that we have further introduced a structure $L(\boldsymbol{x}) = f_\ell(\boldsymbol{A}\boldsymbol{x})$ from the simple sparse estimation problem (1.3). This is useful in decoupling the property of the loss function f_ℓ from that of the design matrix $\boldsymbol{A}$. In a machine learning problem, it is easy to discuss properties of the loss function (because we choose it) but we have to live with whatever property possessed by the design matrix (the data matrix). For notational convenience we assume that for $\eta > 0$, $\eta\phi_\lambda(\boldsymbol{x}) = \phi_{\lambda\eta}(\boldsymbol{x})$; for example, see the ℓ_1-regularizer (1.8).

The dual problem of (P) can be written as the following minimization problem:

$$\text{(D)} \qquad \min_{\boldsymbol{\alpha}\in\mathbb{R}^m,\boldsymbol{v}\in\mathbb{R}^n} \quad f_\ell^*(-\boldsymbol{\alpha}) + \phi_\lambda^*(\boldsymbol{v}), \tag{1.19}$$

$$\text{subject to} \quad \boldsymbol{v} = \boldsymbol{A}^T\boldsymbol{\alpha}, \tag{1.20}$$

where f_ℓ^* and ϕ_λ^* are the convex conjugate functions of f_ℓ and ϕ_λ, respectively.

Let η be a nonnegative real number. The *augmented Lagrangian* (AL) function $J_\eta(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{x})$ is written as follows:

$$J_\eta(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{x}) := f_\ell^*(-\boldsymbol{\alpha}) + \phi_\lambda^*(\boldsymbol{v}) + \langle \boldsymbol{x},\, \boldsymbol{A}^T\boldsymbol{\alpha} - \boldsymbol{v}\rangle + \frac{\eta}{2}\|\boldsymbol{A}^T\boldsymbol{\alpha} - \boldsymbol{v}\|^2. \tag{1.21}$$

Note that the AL function is reduced to the ordinary Lagrangian if $\eta = 0$; the primal variable $\boldsymbol{x}$ appears in the AL function (1.21) as a Lagrangian multiplier vector; it is easy to verify that $\min_{\boldsymbol{\alpha},\boldsymbol{v}} J_0(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{x})$ gives the (sign-inverted) primal objective function (1.18).

Similar to the proximal minimization approach discussed in the previous section, we choose a sequence of positive step-size parameters $\eta_0, \eta_1, \ldots$, and an initial Lagrangian multiplier $\boldsymbol{x}^0$. At every iteration, the DAL algorithm minimizes the AL function $J_{\eta_t}(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{x}^t)$ (1.21) with respect to $(\boldsymbol{\alpha}, \boldsymbol{v})$ and the minimizer $(\boldsymbol{\alpha}^{t+1}, \boldsymbol{v}^{t+1})$ is used to update the Lagrangian multiplier $\boldsymbol{x}^t$ as follows:

$$(\boldsymbol{\alpha}^{t+1}, \boldsymbol{v}^{t+1}) := \operatorname*{argmin}_{\boldsymbol{\alpha},\boldsymbol{v}} J_{\eta_t}(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{x}^t), \tag{1.22}$$

$$\boldsymbol{x}^{t+1} := \boldsymbol{x}^t + \eta_t(\boldsymbol{A}^T\boldsymbol{\alpha}^{t+1} - \boldsymbol{v}^{t+1}). \tag{1.23}$$

Intuitively speaking, we minimize an inner objective (1.22) and update (1.23) the Lagrangian multiplier $\boldsymbol{x}^t$ proportionally to the violation of the equality constraint (1.20). In fact, it can be shown that the direction $(\boldsymbol{A}^T\boldsymbol{\alpha}^{t+1} - \boldsymbol{v}^{t+1})$ is the negative gradient direction of the differentiable auxiliary function $f_{\eta_t}(\boldsymbol{x}) := -\min_{\boldsymbol{\alpha},\boldsymbol{v}} J_{\eta_t}(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{x})$, which coincides with $f(\boldsymbol{x})$ at the optimum; see Bertsekas (1982).

Note that the terms in the AL function (1.21) that involve $\boldsymbol{v}$ are linear, quadratic, and the convex conjugate of the regularizer ϕ_λ. Accordingly, by defining *Moreau's envelope function* (see Appendix 1.B and also Moreau (1965); Rockafellar (1970)) Φ_λ^* as

$$\Phi_\lambda^*(\boldsymbol{y}) := \min_{\boldsymbol{y}' \in \mathbb{R}^n} \left(\phi_\lambda^*(\boldsymbol{y}') + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{y}'\|^2\right), \tag{1.24}$$

we can rewrite the update equations (1.22) and (1.23) as follows:

$$\boldsymbol{\alpha}^{t+1} = \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^m} \underbrace{\left(f_\ell^*(-\boldsymbol{\alpha}) + \frac{1}{\eta_t}\Phi_{\lambda\eta_t}^*(\boldsymbol{x}^t + \eta_t\boldsymbol{A}^T\boldsymbol{\alpha})\right)}_{=:\varphi_t(\boldsymbol{\alpha})}, \tag{1.25}$$

$$\boldsymbol{x}^{t+1} := \operatorname{prox}_{\phi_{\lambda\eta_t}}\left(\boldsymbol{x}^t + \eta_t\boldsymbol{A}^T\boldsymbol{\alpha}^{t+1}\right), \tag{1.26}$$

where we used the identity $\operatorname{prox}_f(\boldsymbol{x}) + \operatorname{prox}_{f^*}(\boldsymbol{x}) = \boldsymbol{x}$ (see Appendix 1.B).

See Tomioka and Sugiyama (2009); Tomioka et al. (2010a) for the derivation.

1.4.2 DAL as a primal proximal minimization

The following proposition states that the DAL algorithm is equivalent to the proximal minimization algorithm in the primal (and thus the algorithm is stable for any positive step-size η_t); see also Table 1.2.

Proposition 1.3. *The iteration* (1.25)-(1.26) *is equivalent to the proximal minimization algorithm* (1.15) *on the primal problem (P).*

Proof. The proximal minimization algorithm for the problem (1.18) is written as follows:

$$\begin{aligned}\boldsymbol{x}^{t+1} &:= \mathop{\mathrm{argmin}}_{\boldsymbol{x}\in\mathbb{R}^n}\left(f_\ell(\boldsymbol{A}\boldsymbol{x}) + \phi_\lambda(\boldsymbol{x}) + \frac{1}{2\eta_t}\|\boldsymbol{x}-\boldsymbol{x}^t\|^2\right)\\ &= \mathop{\mathrm{argmin}}_{\boldsymbol{x}\in\mathbb{R}^n}\left(f_\ell(\boldsymbol{A}\boldsymbol{x}) + \frac{1}{\eta_t}\left(\phi_{\lambda\eta_t}(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^t\|^2\right)\right).\end{aligned}$$

Now we define

$$\Phi_\lambda(\boldsymbol{x};\boldsymbol{x}_t) := \phi_\lambda(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^t\|^2 \tag{1.27}$$

and use the Fenchel duality to obtain

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\left(f_\ell(\boldsymbol{A}\boldsymbol{x}) + \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{x};\boldsymbol{x}^t)\right) = \max_{\boldsymbol{\alpha}\in\mathbb{R}^m}\left(-f_\ell^*(-\boldsymbol{\alpha}) - \frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\eta_t\boldsymbol{A}^T\boldsymbol{\alpha};\boldsymbol{x}^t)\right), \tag{1.28}$$

where f_ℓ^* and $\Phi_\lambda^*(\cdot;\boldsymbol{x}^t)$ are the convex conjugate functions of f_ℓ and $\Phi_\lambda(\cdot;\boldsymbol{x}^t)$, respectively. Here, since $\Phi_\lambda(\cdot;\boldsymbol{x}^t)$ is a sum of two convex functions, its convex conjugate is the infimal convolution (see Appendix 1.A) of the convex conjugates, i.e.

$$\Phi_\lambda^*(\boldsymbol{y};\boldsymbol{x}^t) = \inf_{\tilde{\boldsymbol{v}}\in\mathbb{R}^n}\left(\phi_\lambda^*(\tilde{\boldsymbol{v}}) + \frac{1}{2}\|\boldsymbol{y}-\tilde{\boldsymbol{v}}\|^2 + \langle\boldsymbol{y}-\tilde{\boldsymbol{v}},\,\boldsymbol{x}^t\rangle\right). \tag{1.29}$$

Since, $\Phi_\lambda^*(\boldsymbol{y};\boldsymbol{x}^t) = \Phi_\lambda^*(\boldsymbol{x}^t+\boldsymbol{y};\boldsymbol{0}) = \Phi_\lambda^*(\boldsymbol{x}^t+\boldsymbol{y})$ ignoring a constant term that does not depend on $\boldsymbol{y}$, we have the inner minimization problem (1.25). In order to obtain the update equation (1.26), we turn back to the Fenchel duality theorem and notice that the minimizer $\boldsymbol{x}^{t+1}$ in the left-hand side of Equation (1.28) satisfies

$$\boldsymbol{x}^{t+1} \in \partial_{\boldsymbol{y}}\Phi^*_{\lambda\eta_t}(\boldsymbol{y};\boldsymbol{x}^t)|_{\boldsymbol{y}=\eta_t\boldsymbol{A}^T\boldsymbol{\alpha}^{t+1}}.$$

Since $\Phi_\lambda^*(\boldsymbol{y};\boldsymbol{x}^t)$ is Moreau's envelope function of ϕ_λ^* (ignoring constants), it

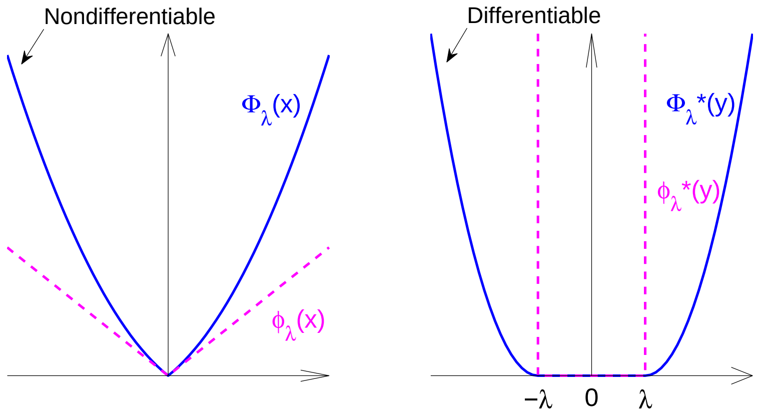

Figure 1.1: Comparison of $\Phi_\lambda(x;0)$ (left) and $\Phi^*_\lambda(y;0)$ (right) for the one-dimensional ℓ_1-regularizer $\phi_\lambda(x) = \lambda|x|$.

is differentiable and the derivative $\nabla_{\boldsymbol{y}} \Phi^*_{\lambda\eta_t}(\eta_t \boldsymbol{A}^T \boldsymbol{\alpha}^{t+1}; \boldsymbol{x}^t)$ is given as follows (see Appendix 1.B):

$$\begin{aligned}\boldsymbol{x}^{t+1} &= \nabla_{\boldsymbol{y}} \Phi^*_{\lambda\eta_t}(\eta_t \boldsymbol{A}^T \boldsymbol{\alpha}^{t+1}; \boldsymbol{x}^t)\\ &= \nabla_{\boldsymbol{y}} \Phi^*_{\lambda\eta_t}(\boldsymbol{x}^t + \eta_t \boldsymbol{A}^T \boldsymbol{\alpha}^{t+1}; \boldsymbol{0}) = \mathrm{prox}_{\phi_{\lambda\eta_t}}(\boldsymbol{x}^t + \eta_t \boldsymbol{A}^T \boldsymbol{\alpha}^{t+1}),\end{aligned}$$

from which we have the update equation (1.26). □

The equivalence of proximal minimization and augmented Lagrangian we have shown above is not novel and it can be found for example in Rockafellar (1976b); Ibaraki et al. (1992). However the above derivation can easily be generalized to the case when the loss function f_ℓ is not differentiable (Suzuki and Tomioka, 2010).

It is worth noting that $\Phi_\lambda(\cdot; \boldsymbol{x}^t)$ is not differentiable but $\Phi^*_\lambda(\cdot; \boldsymbol{x}^t)$ is. See Figure 1.1 for a schematic illustration of the case of one-dimensional ℓ_1-regularizer. Both the ℓ_1-regularizer ϕ_λ and its convex conjugate ϕ^*_λ are nondifferentiable at some points. The function $\Phi_\lambda(\boldsymbol{x}) := \Phi_\lambda(\boldsymbol{x}; \boldsymbol{0})$ is obtained by adding a quadratic proximity term to ϕ_λ; see Equation (1.27). Although Φ_λ is still nondifferentiable, its convex conjugate Φ^*_λ is differentiable due to the infimal convolution operator (see Appendix 1.A) with the proximity term; see Equation (1.29).

The differentiability of Moreau's envelope function Φ^*_λ makes the DAL approach (1.25)-(1.26) computationally efficient. At every step, we minimize a differentiable inner objective (1.25) and use the minimizer to compute the update step (1.26).

1.4.3 Exemplary instance: ℓ_1-regularizer

In order to understand the efficiency of minimizing the inner objective (1.25), let us consider the simplest sparse estimation problem: the ℓ_1-regularization.

For the ℓ_1-regularizer, $\phi_\lambda(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$, the update equations (1.25) and (1.26) can be rewritten as follows:

$$\boldsymbol{\alpha}^{t+1} = \underset{\boldsymbol{\alpha}\in\mathbb{R}^m}{\operatorname{argmin}} \underbrace{\left(f_\ell^*(-\boldsymbol{\alpha}) + \frac{1}{2\eta_t}\left\|\operatorname{prox}_{\lambda\eta_t}^{\ell_1}(\boldsymbol{x}^t + \eta_t \boldsymbol{A}^T\boldsymbol{\alpha})\right\|^2\right)}_{=:\varphi_t(\boldsymbol{\alpha})}, \tag{1.30}$$

$$\boldsymbol{x}^{t+1} = \operatorname{prox}_{\lambda\eta_t}^{\ell_1}\left(\boldsymbol{x}^t + \eta_t \boldsymbol{A}^T\boldsymbol{\alpha}^{t+1}\right), \tag{1.31}$$

where $\operatorname{prox}_\lambda^{\ell_1}$ is the soft-threshold function (1.9); see Tomioka and Sugiyama (2009); Tomioka et al. (2010a) for the derivation.

Note that the second term in the inner objective function $\varphi_t(\boldsymbol{\alpha})$ (1.30) is the squared sum of n one-dimensional soft-thresholds. Thus we only need to compute the sum over the active components $\mathcal{J}^+ := \{j : |x_j^t(\boldsymbol{\alpha})| > \lambda\eta_t\}$ where $\boldsymbol{x}^t(\boldsymbol{\alpha}) := \boldsymbol{x}^t + \eta_t\boldsymbol{A}^T\boldsymbol{\alpha}$. In fact,

$$\left\|\operatorname{prox}_{\lambda\eta_t}^{\ell_1}(\boldsymbol{x}^t(\boldsymbol{\alpha}))\right\|^2 = \sum_{j=1}^n (\operatorname{prox}_{\lambda\eta_t}^{\ell_1}(x_j^t(\boldsymbol{\alpha})))^2 = \sum_{j\in\mathcal{J}^+} (\operatorname{prox}_{\lambda\eta_t}^{\ell_1}(x_j^t(\boldsymbol{\alpha})))^2.$$

Note that the flat area in the plot of $\Phi_\lambda^*(y)$ in Figure 1.1 corresponds to an inactive component.

Moreover, the gradient and the Hessian of $\varphi_t(\boldsymbol{\alpha})$ can be computed as follows:

$$\begin{aligned}\nabla\varphi_t(\boldsymbol{\alpha}) &= -\nabla f_\ell^*(-\boldsymbol{\alpha}) + \boldsymbol{A}\operatorname{prox}_{\lambda\eta_t}^{\ell_1}(\boldsymbol{x}^t + \eta_t\boldsymbol{A}^T\boldsymbol{\alpha}),\\ \nabla^2\varphi_t(\boldsymbol{\alpha}) &= \nabla^2 f_\ell^*(-\boldsymbol{\alpha}) + \eta_t\boldsymbol{A}_+\boldsymbol{A}_+^T,\end{aligned}$$

where $\boldsymbol{A}_+$ is the sub-matrix of $\boldsymbol{A}$ that consists of columns of $\boldsymbol{A}$ that corresponds to the active components $\mathcal{J}^+$. Again, notice that only the active components enter the computation of the gradient and the Hessian.

Looking at Figure 1.1 carefully, one might wonder what happens if the minimizer $\boldsymbol{\alpha}^{t+1}$ lands on a point where $\Phi_\lambda^*(\boldsymbol{y})$ starts to diverge from $\phi_\lambda^*(\boldsymbol{y})$ ($y = -\lambda, \lambda$ in Figure 1.1). In fact, the second derivative of Φ_λ^* is discontinuous on such a point. Nevertheless, we can show that such an event is rare as in the following theorem.

Theorem 1.4. *Assume the regularizer $\phi_\lambda(\boldsymbol{x}) = \lambda\sum_{j=1}^n |x_j|$ (ℓ_1-regularizer). A minimizer $\boldsymbol{x}^*$ of the objective* (1.18) *has no component located exactly at the threshold λ for most λ in the sense that it can be avoided by an arbitrary small perturbation of λ.*

Proof. Optimality condition for the objective (1.18) with the ℓ_1-regularizer can be written as follows:

$$\boldsymbol{x}^* = \mathrm{prox}_\lambda^{\ell_1}(\boldsymbol{x}^* + \boldsymbol{v}^*), \qquad \boldsymbol{v}^* = -\boldsymbol{A}^T \nabla f_\ell(\boldsymbol{A}\boldsymbol{x}^*),$$

which implies $\|\boldsymbol{v}\|_\infty \leq \lambda$ and the complementary slackness conditions

$$x_j \geq 0 \quad \text{if} \quad v_j = \lambda, \tag{1.32a}$$

$$x_j = 0 \quad \text{if} \quad -\lambda < v_j < \lambda, \tag{1.32b}$$

$$x_j \leq 0 \quad \text{if} \quad v_j = -\lambda, \tag{1.32c}$$

for all $j = 1, \ldots, n$. Since the event $x_j = 0$ and $v_j = -\lambda$ or $x_j = 0$ and $v_j = \lambda$ can be avoided by an arbitrary small perturbation of λ for a generic design matrix $\boldsymbol{A}$ and a differentiable loss function f_ℓ, either $x_j^* + v_j^* > \lambda$ (1.32a), $-\lambda < x_j^* + v_j^* < \lambda$ (1.32b), or $x_j^* + v_j^* < -\lambda$ (1.32c) holds, which concludes the proof. □

The above theorem guarantees that the inner objective (1.30) behaves like a twice differentiable function around the optimum for a generic choice of λ and $\boldsymbol{A}$. The theorem can immediately be generalized to the group lasso regularizer (1.10) and the trace-norm regularizer (1.12) by appropriately defining the complementary slackness conditions (1.32a)–(1.32c).

1.4.4 Why do we apply the AL method to the dual?

One reason for applying the AL method to the dual problem (D) is that some loss functions are only strongly convex in the dual; e.g., the logistic loss, which is not strongly convex, becomes strongly convex by taking the convex conjugate; in general loss functions with Lipschitz continuous gradients become strongly convex in the dual; see also Section 1.4.5.

Another reason is that the inner objective function does not have the sparsity we discussed in Section 1.4.3 when the AL method is applied to the primal. In fact, applying the AL method to the primal problem (P) is equivalent to applying the proximal minimization algorithm to the dual problem (D). Therefore, for the ℓ_1-case, the "regularizer" $\phi_\lambda(\boldsymbol{x})$ is defined as follows:

$$\phi_\lambda(\boldsymbol{x}) := (\phi_\lambda^{\ell_1})^*(\boldsymbol{x}) = \begin{cases} 0 & (\text{if } \|\boldsymbol{x}\|_\infty \leq \lambda), \\ +\infty & (\text{otherwise}), \end{cases}$$

which is the convex conjugate of the ℓ_1-regularizer $\phi_\lambda^{\ell_1}$. Adding a quadratic proximity term, we obtain Φ_λ. By taking the convex conjugate of ϕ_λ and Φ_λ, we obtain the ℓ_1-regularizer $\phi_\lambda^* := \phi_\lambda^{\ell_1}$ and Moreau's envelope function

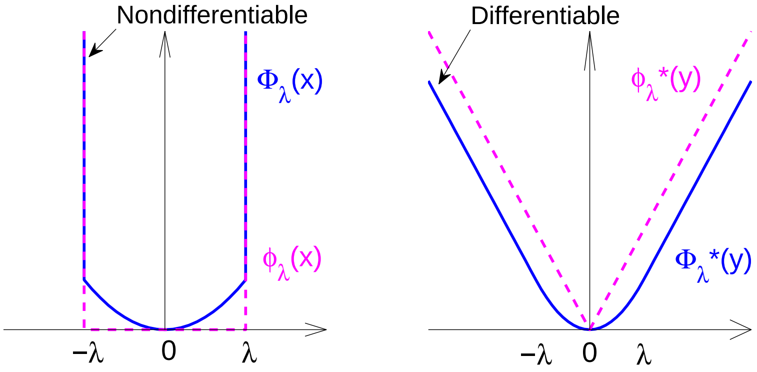

Figure 1.2: Comparison of $\Phi_\lambda(x)$ (left) and $\Phi^*_\lambda(y)$ (right) for the primal application of the AL method to the one-dimensional ℓ_1 problem.

Φ^*_λ of the ℓ_1-regularizer; see Figure 1.2.

Now from Figure 1.2, we can see that the envelope function $\Phi^*_\lambda(y)$ is quadratic for $|y| \leq \lambda$, which corresponds to inactive components, and is linear for $|y| > \lambda$, which corresponds to active components. Thus, we need to compute the terms in the envelope function Φ^*_λ that correspond to both the active and inactive components. Moreover, for the active components the envelope function behaves like a linear function around the minimum, which might be difficult to optimize especially when combined with a loss function that is not strongly convex.

1.4.5 Super-linear convergence of DAL

The asymptotic convergence rate of DAL approach is guaranteed by classic results (see Rockafellar (1976a); Kort and Bertsekas (1976)) under mild conditions even when the inner minimization (1.25) is carried out only approximately. However the condition to stop the inner minimization proposed in Rockafellar (1976a) is often difficult to check in practice. In addition, the analysis in Kort and Bertsekas (1976) assumes strong convexity of the objective. In our setting, the dual objective (1.19) is not necessarily strongly convex as a function of $\boldsymbol{\alpha}$ and $\boldsymbol{v}$; thus we cannot directly apply the result of Kort and Bertsekas (1976) to our problem, though the result is very similar to ours.

Here we provide a non-asymptotic convergence rate of DAL, which generalizes Theorem 1.2 to allow for approximate inner minimization (1.25) with a practical stopping criterion.

Theorem 1.5. *Let $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$ be the sequence generated by the DAL algorithm* (1.25)-(1.26) *and let $\boldsymbol{x}^*$ be a minimizer of the objective function f. Assume the same condition* **(A1)** *as in Theorem 1.2 and in addition assume that the following conditions hold:*

(A2) *The loss function f_ℓ has Lipschitz continuous gradient with modulus $1/\gamma$, i.e.,*

$$\|\nabla f_\ell(\boldsymbol{z}) - \nabla f_\ell(\boldsymbol{z}')\| \leq \frac{1}{\gamma}\|\boldsymbol{z} - \boldsymbol{z}'\| \qquad (\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^m). \tag{1.33}$$

(A3) *The proximity operator corresponding to ϕ_λ can be computed exactly.*

(A4) *The inner minimization* (1.25) *is solved to the following tolerance:*

$$\|\nabla \varphi_t(\boldsymbol{\alpha}^{t+1})\| \leq \sqrt{\frac{\gamma}{\eta_t}}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|,$$

where γ is the constant in assumption **(A2)***.*

Under assumptions **(A1)**-**(A4)***, the following inequality is true:*

$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^{\frac{1+\alpha\sigma\eta_t}{1+2\sigma\eta_t}} \leq \frac{1}{\sqrt{1+2\sigma\eta_t}}\|\boldsymbol{x}^t - \boldsymbol{x}^*\|.$$

That is, $\boldsymbol{x}^t$ converges to $\boldsymbol{x}^*$ super-linearly if $\alpha < 2$ or $\alpha = 2$ and η_t is increasing.

Proof. See Tomioka et al. (2010a). □

Note that the above stopping criterion **(A4)** is computable, since the Lipschitz constant γ only depends on the loss function used and not on the data matrix $\boldsymbol{A}$. Although the constant σ in assumption **(A1)** is difficult to quantify in practice, it is enough to know that it exists, because we do not need σ to compute the stopping criterion **(A4)**. See Tomioka et al. (2010a) for more details.

1.5 Connections

The AL formulation in the dual is connected to various operator theoretic algorithms in the primal. We have already seen that the exact application of DAL corresponds to the proximal point algorithm in the primal (Section 1.4.2). In this section, we show that two well known *operator splitting* algorithms, namely forward-backward splitting and Douglas-Rachford splitting in the primal can be regarded as some approximate computations of the DAL approach. The results in this section are not novel and are based

Table 1.2: Primal-dual correspondence of operator splitting algorithms and augmented Lagrangian algorithms.

	Primal	Dual
Exact	Proximal minimization	Augmented Lagrangian algorithm (Rockafellar, 1976b)
Approximation	Forward-backward splitting	Alternating minimization algorithm (Tseng, 1991)
	Douglas-Rachford splitting	Alternating direction method of multipliers (Gabay and Mercier, 1976)

on Lions and Mercier (1979); Eckstein and Bertsekas (1992); Tseng (1991); see also recent reviews in Yin et al. (2008); Setzer (2010); Combettes and Pesquet (2010). The methods we discuss in this section are summarized in Table 1.2.

Note that these approximations are most valuable when the inner minimization problem (1.22) is not easy to minimize. In Goldstein and Osher (2009), an approximate AL method was applied to a *structured* sparse estimation problem, namely the total-variation denoising.

In this section we use the notation $L(\boldsymbol{x}) = f_\ell(\boldsymbol{A}\boldsymbol{x})$ for simplicity, since the discussions does not require the separation between the loss function and the design matrix as in Section 1.4.

1.5.1 Forward-backward splitting

When the loss function L is differentiable, replacing the inner minimization (1.22) by the following sequential minimization steps:

$$\boldsymbol{\alpha}^{t+1} = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^m} J_0(\boldsymbol{\alpha}, \boldsymbol{v}^t; \boldsymbol{x}^t), \tag{1.34}$$

$$\boldsymbol{v}^{t+1} = \operatorname*{argmin}_{\boldsymbol{v}\in\mathbb{R}^n} J_{\eta_t}(\boldsymbol{\alpha}^{t+1}, \boldsymbol{v}; \boldsymbol{x}^t) \tag{1.35}$$

gives the forward-backward splitting (FBS) algorithm (Lions and Mercier, 1979; Combettes and Wajs, 2005; Combettes and Pesquet, 2010):

$$\boldsymbol{x}^{t+1} = \operatorname{prox}_{\phi_{\lambda\eta_t}}\left(\boldsymbol{x}^t - \eta_t \nabla L(\boldsymbol{x}^t)\right). \tag{1.36}$$

Note that in the first step (1.34), the ordinary Lagrangian ($\eta = 0$) is used and the augmented Lagrangian is only used in the second step (1.35). The above sequential procedure is proposed in Han and Lou (1988) and analyzed in Tseng (1991) under the name "alternating minimization algorithm".

The FBS algorithm was proposed in the context of finding a zero of the operator equation (1.2). When the operator A is single valued, the operator equation (1.2) implies

$$(I + \eta B)(\boldsymbol{x}) \ni (I - \eta A)(\boldsymbol{x}).$$

This motivates us to use the following iteration

$$\boldsymbol{x}^{t+1} = (I + \eta B)^{-1}(I - \eta A)(\boldsymbol{x}^t).$$

The above iteration converge to the solution of the operator equation (1.2) if A is Lipschitz continuous and the step-size η is small enough (see Lions and Mercier (1979); Combettes and Wajs (2005)). The iteration (1.36) is obtained by identifying $A = \nabla L$ and $B = \partial\phi_\lambda$. Intuitively, the FBS algorithm takes an *explicit* (forward) gradient step with respect to the differentiable term L and then takes an *implicit* (backward) gradient step (1.17) with respect to the nondifferentiable term ϕ_λ.

The FBS algorithm is also known as the iterative shrinkage/thresholding (IST) algorithm (see Figueiredo and Nowak (2003); Daubechies et al. (2004); Figueiredo et al. (2007); Wright et al. (2009); Beck and Teboulle (2009) and the references therein). The FBS algorithm converges as fast as the gradient descent on the loss term in problem (1.3). For example, when the loss term has Lipschitz continuous gradient and strongly convex, it converges linearly (Tseng, 1991). However, this is rarely the case in sparse estimation because typically the number of unknowns n is larger than the number of observations m. Beck and Teboulle (2009) proved that FBS converges as $O(1/k)$ without the strong convexity assumption. However, since the Lipschitz constant depends on the design matrix $\boldsymbol{A}$, it is difficult to quantify it for a machine learning problem. Nesterov (2007) and Beck and Teboulle (2009) proposed accelerated IST algorithms that converge as $O(1/k^2)$, which is also optimal under the first order black-box model (Nesterov, 2007). The connection between the accelerated IST algorithm and the operator splitting framework is unknown.

1.5.2 Douglas-Rachford splitting

Another commonly used approximation to minimize the inner objective function (1.22) is to perform minimization with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{v}$ alternately, which is called the *alternating direction method of multipliers* (Gabay and Mercier, 1976). This approach is known to be equivalent to the Douglas-

Rachford splitting (DRS) algorithm (Douglas and Rachford, 1956; Lions and Mercier, 1979; Eckstein and Bertsekas, 1992; Combettes and Pesquet, 2010) when the proximity parameter η_t is chosen to be constant $\eta_t = \eta$.

Similar to the FBS algorithm, DRS algorithm splits the operator equation (1.2) as follows:

$$(I + \eta B)(\boldsymbol{x}) \ni \boldsymbol{x} - \eta \boldsymbol{y}, \qquad (I + \eta A)(\boldsymbol{x}) \ni \boldsymbol{x} + \eta \boldsymbol{y}.$$

Accordingly, starting from some appropriate initial point $(\boldsymbol{x}^0, \boldsymbol{y}^0)$, the DRS algorithm performs the following iteration:

$$\left(\boldsymbol{x}^{t+1}, \eta \boldsymbol{y}^{t+1}\right) = \mathrm{decomp}_{\eta A}\left((I + \eta B)^{-1}(\boldsymbol{x}^t - \eta \boldsymbol{y}^t) + \eta \boldsymbol{y}^t\right),$$

where with a slight abuse of notation, we denote by $(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{decomp}_A(\boldsymbol{z})$ the decomposition $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{z}$ with $\boldsymbol{x} = (I + A)^{-1}(\boldsymbol{z})$; note that this implies $\boldsymbol{y} \in A(\boldsymbol{x})$; see the original definition (1.7).

Turning back to the DAL algorithm (1.22)-(1.23), due to the symmetry between $\boldsymbol{\alpha}$ and $\boldsymbol{v}$, there are two ways to convert the DAL algorithm to a DRS algorithm. First, by replacing the inner minimization (1.22) by the following steps,

$$\boldsymbol{v}^{t+1} = \underset{\boldsymbol{v} \in \mathbb{R}^n}{\mathrm{argmin}}\, J_\eta(\boldsymbol{\alpha}^t, \boldsymbol{v}; \boldsymbol{x}^t), \qquad \boldsymbol{\alpha}^{t+1} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^m}{\mathrm{argmin}}\, J_\eta(\boldsymbol{\alpha}, \boldsymbol{v}^{t+1}; \boldsymbol{x}^t),$$

we obtain the (primal) DRS algorithm:

$$\left(\boldsymbol{x}^{t+1}, -\eta \boldsymbol{A}^T \boldsymbol{\alpha}^{t+1}\right) = \mathrm{decomp}_{\eta L}\left(\mathrm{prox}_{\phi_{\lambda\eta}}\left(\boldsymbol{x}^t + \eta \boldsymbol{A}^T \boldsymbol{\alpha}^t\right) - \eta \boldsymbol{A}^T \boldsymbol{\alpha}^t\right), \tag{1.37}$$

where $(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{decomp}_{\eta L}(\boldsymbol{z})$ denotes Moreau's decomposition (1.7). We can identify $A = \partial L$ and $B = \partial \phi_\lambda$ in update equation (1.37). This version of DRS ("regularizer inside, loss outside") was considered in Combettes and Pesquet (2007) for image denoising with non-Gaussian likelihood models. When the loss function L is differentiable, update equation (1.37) can be simplified as follows:

$$\boldsymbol{x}^{t+1} = \mathrm{prox}_{\eta L}\left(\mathrm{prox}_{\phi_{\lambda\eta}}(\boldsymbol{x}^t - \eta \nabla L(\boldsymbol{x}^t)) + \eta \nabla L(\boldsymbol{x}^t)\right),$$

which more resembles the FBS iteration (1.36).

On the other hand, by replacing the inner minimization (1.22) by the following steps,

$$\begin{aligned}\boldsymbol{\alpha}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^m} J_\eta(\boldsymbol{\alpha}, \boldsymbol{v}^t; \boldsymbol{x}^t),\\ \boldsymbol{v}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{v}\in\mathbb{R}^n} J_\eta(\boldsymbol{\alpha}^{t+1}, \boldsymbol{v}; \boldsymbol{x}^t),\end{aligned}$$

we obtain another (primal) DRS algorithm:

$$\left(\boldsymbol{x}^{t+1}, \eta\boldsymbol{v}^{t+1}\right) = \operatorname{decomp}_{\phi_{\lambda\eta}}\left(\operatorname{prox}_{\eta L}(\boldsymbol{x}^t - \eta\boldsymbol{v}^t) + \eta\boldsymbol{v}^t\right). \tag{1.38}$$

Here, we can identify $A = \partial\phi_\lambda$ and $B = \partial L$ in update equation (1.38). This version of DRS ("loss inside, regularizer outside") was proposed in Goldstein and Osher (2009) to as an alternating direction method for the total-variation denoising problem (1.5).

Each step of DRS is a firmly nonexpansive mapping, and thus DRS is unconditionally stable (Lions and Mercier, 1979), whereas the stability of FBS depends on the choice of the proximity parameter η. Moreover, DRS can be applied in both ways (see update equations (1.37) and (1.38)). In other words, both the loss function L and the regularizer ϕ_λ may be nondifferentiable, whereas FBS assumes that the loss L is differentiable. However, this also means that both proximity operators need to be implemented for DRS, whereas FBS requires only one of them (Combettes and Pesquet, 2010).

1.6 Application

In this section, we demonstrate that the trace norm regularizer (1.12) can be used to learn features from multiple sources and combine them in an optimal way in a single optimization problem. We also demonstrate that DAL can efficiently optimize the associated minimization problem.

1.6.1 Problem setting

The problem we solve is a classification problem with multiple matrix-valued inputs (Tomioka et al., 2010b), namely

$$\min_{\substack{\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(K)},\\ b\in\mathbb{R}}} \sum_{i=1}^{m} \ell\left(\textstyle\sum_{k=1}^{K}\langle \boldsymbol{X}_i^{(k)}, \boldsymbol{W}^{(k)}\rangle + b, y_i\right) + \lambda\sum_{k=1}^{K}\|\boldsymbol{W}^{(k)}\|_*, \tag{1.39}$$

where the loss function ℓ is the logistic loss function

$$\ell(z, y) = \log(1 + \exp(-yz)), \tag{1.40}$$

and $\|\cdot\|_*$ denotes the trace norm (1.12).

By defining

$$\begin{aligned}
\boldsymbol{x} &= \left(\mathrm{vec}(\boldsymbol{W}^{(1)})^T, \ldots, \mathrm{vec}(\boldsymbol{W}^{(K)})^T, b\right)^T, \\
f_\ell(\boldsymbol{z}) &= \sum_{i=1}^{m} \ell(z_i, y_i), \\
\boldsymbol{A} &: \text{ an } m \times n \text{ matrix whose } i\text{th row is given as} \\
\boldsymbol{A}_i &= \left(\mathrm{vec}(\boldsymbol{X}_i^{(1)})^T, \ldots, \mathrm{vec}(\boldsymbol{X}_i^{(K)})^T, 1\right), \\
\phi_\lambda(\boldsymbol{x}) &= \lambda \sum_{k=1}^{K} \|\boldsymbol{W}^{(k)}\|_*,
\end{aligned}$$

we can see that problem (1.39) is a special case of problem (1.18).

As a concrete example, we take a data-set from a real brain-computer interface (BCI) experiment, where the task is to predict whether the upcoming voluntary finger movement is either right or left hand from the electroencephalography (EEG) measurements (Blankertz et al., 2002). The data-set is made publicly available through the BCI competition 2003 (data-set IV) (Blankertz et al., 2004). More specifically, the data-set consists of short segments of 28 channel multivariate signal of length 50 (500 ms long at 100 Hz sampling). The training set consists of $m = 316$ input segments (159 left and 157 right) and we tested the classifier on a separate test-set consisting of 100 test segments.

Preprocessing

Following the preprocessing used in Tomioka and Müller (2010), we compute three matrices from each segment. The first matrix is 28×50 and is obtained directly from the original signal by low-pass filtering at 20Hz. The second matrix is 28×28 and is derived by computing the covariance between the channels in the frequency band 7-15Hz (known as the α-band). Finally, the third matrix is 28×28 and is computed similarly to the second matrix in the frequency band 15-30Hz (known as the β-band). The total number of unknown variables is $n = 2969$.

We chose 20 log-linearly separated values of the regularization constant λ from 10 to 0.001. The proximity parameter is increased geometrically as $\eta_t = 1, 2, 4, 8, \ldots$; note that after 22 iterations it was as large as $2^{21} \simeq 2.1 \times 10^6$, which shows that DAL is stable across wide range of η_t. The Lipschitz constant γ (see assumption **(A2)** in Theorem 1.5) for the logistic loss (1.40) is $\gamma = 4$. We used the Newton method for the inner minimization

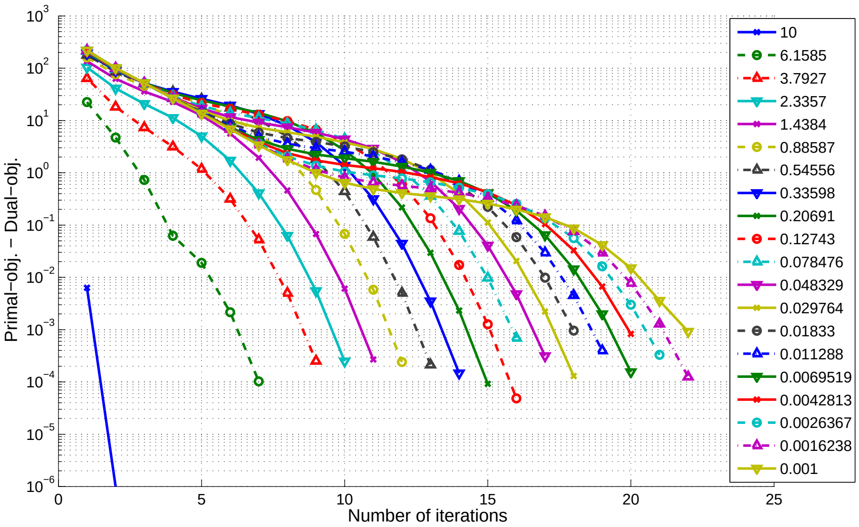

Figure 1.3: Convergence of DAL algorithm applied to a classification problem in BCI. The duality gap is plotted against the number of iterations. Each curve corresponds to different regularization constant λ shown on the right. Note that no warm start is used. Each iteration consumed roughly 1.2 seconds.

problem (1.25). We implemented DAL in Matlab [6]. Each optimization was terminated when the duality gap fell below 10^{-3}; see Section 1.C.

1.6.2 Results

Figure 1.3 shows the sequence of the duality gap obtained by running the DAL algorithm on 20 different values of regularization constant λ against the number of iterations. Note that the vertical axis is logarithmically scaled. We can see that the convergence of DAL becomes faster as the iteration proceeds; i.e., it converges super-linearly. Each iteration consumed roughly 1.2 seconds on a Linux server with two 3.33 GHz Xeon processors, and the computation for 20 values of the regularization constant λ took about 350 seconds. Note that applying a simple warm start can significantly speedup the computation (about 70% reduction) but it is not used here because we are interested in the basic behaviour of the DAL algorithm.

Figure 1.4 shows the singular-value spectra of the coefficient matrices $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$, and $\boldsymbol{W}^{(3)}$ obtained at the regularization constant $\lambda = 0.5456$,

[6] The code is available from `http://www.ibis.t.u-tokyo.ac.jp/RyotaTomioka/Softwares`.

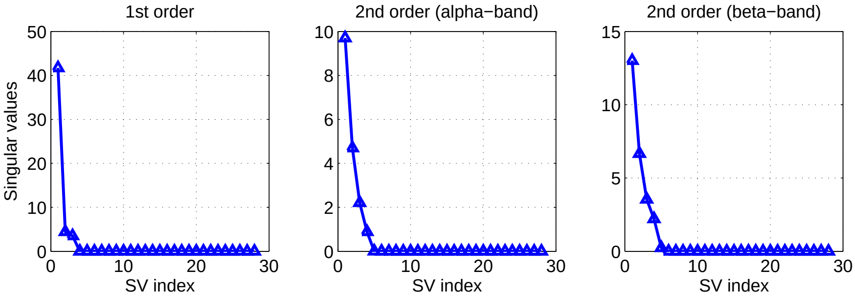

Figure 1.4: Singular-value spectra of $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$, and $\boldsymbol{W}^{(3)}$, which correspond to the first-order component, second order (alpha) component, and second order (beta) component, respectively, obtained by solving optimization problem (1.39) at $\lambda = 0.5456$.

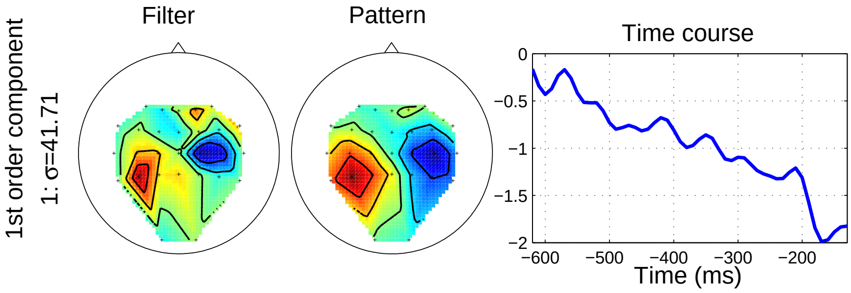

Figure 1.5: The visualization of the left singular-vector ("filter") and the right singular-vector ("time course") corresponding to the largest singular-value of $\boldsymbol{W}^{(1)}$. Both "filter" and "pattern" are shown topographically on a head seen from above. The "pattern" shows the typical activity captured by the "filter". See Tomioka and Müller (2010) for more details.

which achieved the highest test accuracy 85%. The classifier has selected three components from the first data source (first-order component), four components from the second data source (second-order (α-band) component), and five components from the third data source (second-order (β-band) component). From the magnitude of the singular-values, it seems that the first-order component and the β-component are the most important for the classification, whereas the contribution of the α-component is less prominent; see Tomioka and Müller (2010).

Within each data source, the trace norm regularization automatically learns feature extractors. Figure 1.5 visualizes the spatio-temporal profile of the learned feature extractor that corresponds to the leading singular-value of $\boldsymbol{W}^{(1)}$ in Figure 1.4. The “filter” (left) and the “pattern” (center) visualize the left singular-vector topographically according to the geometry of the EEG sensors. The “time course” (right) shows the right singular-vector as a time-series. Both the “filter” and “pattern” show a clear lateralized bipolar structure. This bipolar structure together with the downward trend in the “time course” is physiologically known as the lateralized readiness potential (or Bereitschaftspotential) (Cui et al., 1999). Note that the “time course” starts 630 ms and ends 130 ms *prior* to the actual movement because the task is to predict the laterality of the movement before it is executed.

1.7 Summary

In this chapter, we have presented the dual augmented-Lagrangian (DAL) algorithm for sparse estimation problems, and discussed its connections to proximal minimization and other operator splitting algorithms.

DAL algorithm is an augmented Lagrangian algorithm (Powell, 1969; Hestenes, 1969; Rockafellar, 1976b; Bertsekas, 1982) applied to the dual of the simple sparse estimation problem (1.3). For this problem, the sparsity of the intermediate solution can effectively be exploited to efficiently solve the inner minimization problem. This link between the sparsity and efficiency distinguishes DAL from other AL algorithms.

We have shown that DAL is equivalent to the proximal minimization algorithm in the primal, which enabled us to rigorously analyze the convergence rate of DAL through the proximal minimization framework. We have shown that DAL converges superlinearly even in case of inexact inner minimization. Importantly, the stopping criterion we used can be computed in practice; this is because we have separated the loss function f_ℓ from the design matrix $\boldsymbol{A}$; see Section 1.4.1.

The *structured* sparse estimation problem (1.4) can also be tackled through augmented Lagrangian algorithms in the primal (see Goldstein and Osher (2009); Lin et al. (2009)). However as we discussed in Section 1.4.4, for these algorithms the inner minimization is not easy to carry out exactly, because the convex conjugate regularizer ϕ_λ^* does not produce a sparse vector through the associated proximity operator.

Currently we are interested in how much the insights we gained about DAL transfers to *approximate* augmented Lagrangian algorithms, e.g., alternating direction method, applied to the primal problem (structured sparse estimation) and the dual problem (simple sparse estimation) and the associated operator splitting methods in their respective dual problems. Application of augmented Lagrangian algorithms to kernel methods is another interesting direction (Suzuki and Tomioka, 2010).

Acknowledgement

We would like to thank Masakazu Kojima and Masao Fukushima for helpful discussions. This work was partially supported by MEXT KAKENHI 22700138, 22700289, and the FIRST program.

Appendix

1.A Infimal convolution

Let $f:\mathbb{R}^n \to \mathbb{R}$ and $g:\mathbb{R}^n \to \mathbb{R}$ be two convex functions, and let f^* and g^* be their convex conjugate functions, respectively; That is,

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \left(\langle \boldsymbol{y}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\right), \quad g^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \left(\langle \boldsymbol{y}, \boldsymbol{x}\rangle - g(\boldsymbol{x})\right).$$

Then,

$$(f+g)^*(\boldsymbol{y}) = \inf_{\boldsymbol{y}'\in\mathbb{R}^n} \big(f^*(\boldsymbol{y}') + g^*(\boldsymbol{y}-\boldsymbol{y}')\big) =: (f^*\Box g^*)(\boldsymbol{y}),$$

where $\Box$ denotes the *infimal convolution.*

See (Rockafellar, 1970, Theorem 16.4) for the proof.

1.B Moreau's theorem

Let $f:\mathbb{R}^n \to \mathbb{R}$ be a convex function, and f^* be its convex conjugate function. Then, for $\boldsymbol{x}\in\mathbb{R}^n$

$$\mathrm{prox}_f(\boldsymbol{x}) + \mathrm{prox}_{f^*}(\boldsymbol{x}) = \boldsymbol{x}. \tag{1.41}$$

Moreover,

$$\widehat{f}(\boldsymbol{x}) + \widehat{f^*}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2, \tag{1.42}$$

where $\widehat{f}$ is Moreau's envelope function of f, namely

$$\widehat{f}(\boldsymbol{x}) = \min_{\boldsymbol{x}'\in\mathbb{R}^n} \left(f(\boldsymbol{x}') + \frac{1}{2}\|\boldsymbol{x}'-\boldsymbol{x}\|^2\right).$$

Furthermore, the envelope function $\widehat{f}$ is differentiable. The gradient is given as follows:

$$\nabla \widehat{f}(\boldsymbol{x}) = \operatorname{prox}_{f^*}(\boldsymbol{x}), \qquad \nabla \widehat{f^*}(\boldsymbol{x}) = \operatorname{prox}_{f}(\boldsymbol{x}).$$

See Moreau (1965) and (Rockafellar, 1970, Theorem 31.5) for the proof. Danskin's theorem (Bertsekas, 1999, Proposition B.25) can also be used to show the result. Note that differentiating both sides of Equation (1.42), we obtain Equation (1.41), which confirms the validity of the above statement.

1.C Computation of the duality gap

We use the same strategy used in Koh et al. (2007); Wright et al. (2009) to compute the duality gap as a stopping criterion for the DAL algorithm.

Let $\bar{\boldsymbol{\alpha}}^t := -\nabla f_\ell(\boldsymbol{A}\boldsymbol{x}^t)$. Note that the vector $\boldsymbol{A}^T\bar{\boldsymbol{\alpha}}^t$ does not necessarily lie in the domain of ϕ_λ^* in the dual problem (1.19). For the trace norm regularization, the domain of ϕ_λ^* is matrices with maximum singular-value equal to or smaller than λ. Thus we define $\tilde{\boldsymbol{\alpha}}^t = \bar{\boldsymbol{\alpha}}^t \min(1, \lambda/\|\boldsymbol{A}^T\bar{\boldsymbol{\alpha}}^t\|)$, where $\|\cdot\|$ is the spectral norm. Notice that $\|\boldsymbol{A}^T\tilde{\boldsymbol{\alpha}}^t\| \leq \lambda$ by construction. We compute the dual objective value as $d(\boldsymbol{x}^t) = -f_\ell^*(-\tilde{\boldsymbol{\alpha}}^t)$. Finally the duality gap Gap^t is obtained as $\mathrm{Gap}^t = f(\boldsymbol{x}^t) - d(\boldsymbol{x}^t)$, where f is the primal objective function (1.18).

References

F. R. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

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

D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999. 2nd edition.

B. Blankertz, G. Curio, and K.-R. Müller. Classifying single trial EEG: Towards brain computer interfacing. In T. G. Diettrich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Inf. Proc. Systems (NIPS 01)*, volume 14, pages 157–164, 2002.

B. Blankertz, K.-R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R.

Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer. The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Trans. Biomed. Eng.*, 51(6):1044–1051, 2004.

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

J.-F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. arXiv:0810.3286, 2008.

E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

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*, 1(4):564–574, 2007.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2010.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

R. Q. Cui, D. Huter, W. Lang, and L. Deecke. Neuroimage of voluntary movement: Topography of the bereitschaftspotential, a 64-channel DC current source density study. *Neuroimage*, 9(1):124–134, 1999.

I. Daubechies, M. Defrise, and C. De Mol. An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint. *Commun. Pur. Appl. Math.*, LVII:1413–1457, 2004.

D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, 1995.

J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82 (2):421–439, 1956.

J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

M. Fazel, H. Hindi, and S. P. Boyd. A Rank Minimization Heuristic with

Application to Minimum Order System Approximation. In *Proc. of the American Control Conference*, 2001.

M. A. T. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 12:906–916, 2003.

M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization-Minimization Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Process.*, 16(12), 2007.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17–40, 1976.

J. Gao, G. Andrew, M. Johnson, and K. Toutanova. A comparative study of parameter estimation methods for statistical natural language processing. In *Annual Meeting of the Association for Computational Linguistics*, volume 45, pages 824–831, 2007.

T. Goldstein and S. Osher. The split Bregman method for L1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

S.-P. Han and G. Lou. A parallel algorithm for a class of convex programs. *SIAM J. Control Optimiz.*, 26(2):345–355, 1988.

M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969.

R. A. Horn and C. R. Johnson. *Topics in matrix analysis.* Cambridge University Press, 1991.

S. Ibaraki, M. Fukushima, and T. Ibaraki. Primal-dual proximal point algorithm for linearly constrained convex programming problems. *Computational Optimization and Applications*, 1(2):207–226, 1992.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523, 2009.

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky. An Interior-Point Method for Large-Scale l-Regularized Least Squares. *IEEE journal of selected topics in signal processing*, 1:606–617, 2007.

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*, 8:1519–1555, 2007.

B. W. Kort and D. P. Bertsekas. Combined primal–dual and penalty methods for convex programming. *SIAM Journal on Control and Optimization*, 14(2):268–294, 1976.

Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of a corrupted low-rank matrices. *Mathematical*

Programming, 2009. submitted.

P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid mr imaging, 2007. *Magn. Reson. Med.*, 58 (6):1182–1195, 2007.

J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la S. M. F.*, 93:273–299, 1965.

R. M. Neal. *Bayesian Learning for Neural Networks.* Springer, New York, 1996.

Y. Nesterov. Gradient methods for minimizing composite objective function. Technical Report 2007/76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007.

A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proc. of the twenty-first International Conference on Machine Learning*, page 78, New York, NY, USA, 2004. ACM.

M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, London, New York, 1969.

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

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976a.

R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. of Oper. Res.*, 1:97–116, 1976b.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

S. Setzer. Operator splittings, Bregman methods and frame shrinkage in image processing. *International Journal of Computer Vision*, 2010.

S. K. Shevade and S. S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-Margin Matrix Factorization. In *Advances in NIPS 17*, pages 1329–1336. MIT Press, Cambridge, MA, 2005.

T. Suzuki and R. Tomioka. SpicyMKL. *Machine Learning*, 2010. Submitted.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Stat.*

Soc. B, 58(1):267–288, 1996.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. Roy. Stat. Soc. B*, 67(1):91–108, 2005.

M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

R. Tomioka and K. Aihara. Classifying Matrices with a Spectral Regularization. In *Proc. of the 24th International Conference on Machine Learning*, pages 895–902. ACM Press, 2007.

R. Tomioka and K.-R. Müller. A regularized discriminative framework for EEG analysis with application to brain-computer interface. *Neuroimage*, 49(1):415–432, 2010.

R. Tomioka and M. Sugiyama. Dual augmented Lagrangian method for efficient sparse reconstruction. *IEEE Signal Processing Letters*, 16(12): 1067–1070, 2009.

R. Tomioka, T. Suzuki, and M. Sugiyama. Super-linear convergence of dual augmented-Lagrangian algorithm for sparsity regularized estimation. Technical report, arXiv:0911.4046v2, 2010a.

R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. A fast augmented Lagrangian algorithm for learning low-rank matrices. In *Proc. of the 27th International Conference on Machine Learning*. Omnipress, 2010b.

P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optimiz.*, 29 (1):119–138, 1991.

J. B Weaver, Y. Xu, D. M. Healy Jr, and L. D. Cromwell. Filtering noise from images with wavelet transforms. *Magnetic Resonance in Medicine*, 21(2):288–295, 1991.

D. Wipf and S. Nagarajan. A new view of automatic relevance determination. In *Advances in NIPS 20*, pages 1625–1632. MIT Press, 2008.

S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse Reconstruction by Separable Approximation. *IEEE Trans. Signal Process.*, 2009.

W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman Iterative Algorithms for L1-Minimization with Applications to Compressed Sensing. *SIAM J. Imaging Sciences*, 1(1):143–168, 2008.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. B*, 68(1):49–67, 2006.

M. Yuan, A. Ekici, Z. Lu, and R. Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *J. Roy. Stat. Soc. B*, 69(3):329–346, 2007.