University of Alberta

FAST GRADIENT ALGORITHMS FOR STRUCTURED SPARSITY

by

Yaoliang Yu

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Statistical Machine Learning

Department of Computing Science

Spring 2014
Edmonton, Alberta

To my *grandpa*.

Abstract

Many machine learning problems can be formulated under the composite minimization framework which usually involves a smooth loss function and a nonsmooth regularizer. A lot of algorithms have thus been proposed and the main focus has been on first order gradient methods, due to their applicability in very large scale application domains. A common requirement of many of these popular gradient algorithms is the access to the proximal map of the regularizer, which unfortunately may not be easily computable in scenarios such as structured sparsity. In this thesis we first identify conditions under which the proximal map of a sum of functions is simply the composition of the proximal map of each individual summand, unifying known and uncover novel results. Next, motivated by the observation that many structured sparse regularizers are merely the sum of simple functions, we consider a linear approximation of the proximal map, resulting in the so-called proximal average. Surprisingly, combining this approximation with fast gradient schemes yields strictly better convergence rates than the usual smoothing strategy, without incurring any overhead. Finally, we propose a generalization of the conditional gradient algorithm which completely abandons the proximal map but requires instead the polar—a significantly cheaper operation in certain matrix applications. We establish its convergence rate and demonstrate its superiority on some matrix problems, including matrix completion, multi-class and multi-task learning, and dictionary learning.

Acknowledgements

Writing a PhD thesis requires tremendous efforts and helps from many people. Luckily, I had the privilege to work with two fantastic supervisors, Dale Schuurmans and Csaba Szepesvári, who have always been encouraging, supportive and understanding. Needless to say, I have learned a lot (not just academically!) from them. I am especially grateful for the academic freedom they gave me. I would also like to thank András György, for serving in my supervisory committee and the many interesting conversations. I greatly appreciate Dr. Salavatipour and Dr. Sacchi for being my thesis examiners, and Dr. Bach for his kindness and his many inspiring work.

Part of this thesis is a result of discussion, exchange and joint work with Xinhua Zhang, a good friend who has generously helped me in many ways. And I am terribly sorry for once falling asleep hence not being able to respond to his knocking my door, even though he desperately needed rest after some 20 hours flight. Particularly, I thank Xinhua and Bob Williamson for inviting me to NICTA. One of the results in this thesis (Theorem 2.5) was in fact (im)proved on the plane to Australia—demonstrating that one can do magical things when "high".

Many of the friends at UofA have made my life in Edmonton more enjoyable. I thank them all for the accompany, in particular my roommate Shunjie Lau for his bearing with my anxiety during job hunting.

Last and of course the most, I thank my family, in particular my wife, Sun Sun, for many years of love and support. The very recent evidence of Sun's impeccable role in my life is her urge, company and fine cooking that have made this thesis possible.

Contents

1 Introduction **1**

1.1 Examples . . . 1
1.2 Subgradient Descent . . . 6
1.3 Proximal Gradient . . . 8
1.4 Proximal Subgradient . . . 12
1.5 Regularized Dual Averaging . . . 13
1.6 Structured Sparsity . . . 15
1.7 Contributions . . . 19

2 Prox-decomposition **22**

2.1 Introduction . . . 22
2.2 Proximal Map . . . 23
2.3 Decomposition . . . 25
2.3.1 A Sufficient Condition . . . 27
2.3.2 No Invariance . . . 29
2.3.3 Scaling Invariance . . . 30
2.3.4 Cone Invariance . . . 36
2.4 Connection with the Representer Theorem . . . 38
2.5 Summary . . . 44

3 Proximal Average Approximation **46**

3.1 Introduction . . . 46
3.2 Problem Formulation . . . 47
3.3 Technical Tools . . . 50
3.4 Theoretical Justification . . . 55
3.5 Comparing to Existing Approaches . . . 56
3.6 Some Refinements . . . 58
3.6.1 Optimal weight . . . 58
3.6.2 De-smoothing . . . 59
3.6.3 Nonsmooth Loss . . . 60

3.6.4 Varying Step Size . . . 60
3.7 Experiments . . . 62
3.8 Summary . . . 63

4 Generalized Conditional Gradient **64**
4.1 Generalized Conditional Gradient . . . 64
4.1.1 General Case . . . 65
4.1.2 Lipschitz Case . . . 70
4.1.3 Convex Case . . . 73
4.1.4 Positively Homogeneous Case . . . 75
4.1.5 Refinements and Comments . . . 80
4.1.6 Examples . . . 81
4.2 Dictionary Learning . . . 83
4.2.1 Convex Relaxation . . . 83
4.2.2 Fixed-Rank Local Optimization . . . 87
4.3 Experimental Results . . . 90
4.3.1 Matrix completion . . . 90
4.3.2 Multi-class and multi-task learning . . . 91
4.4 Summary . . . 93

5 Conclusions and Future Directions **94**

Bibliography **103**

A Constructing Convex Regularizers **104**
A.1 Some Basic Results . . . 105
A.2 Example 1: Sparsity . . . 106
A.2.1 ℓ_p-Norm Regularization . . . 106
A.2.2 Truncation . . . 107
A.3 Example 2: Low Rank . . . 109
A.4 Example 3: Dimensionality Reduction . . . 110

List of Figures

1.1 The zero-one, hinge, and logistic loss, as functions of $y\hat{y}$. 3
1.2 The "magic" of the ℓ_1 norm. 4
1.3 The ℓ_1 norm relaxation of $\|\cdot\|_0$. 5
1.4 The Moreau envelop and the proximal map of the absolute function $|\cdot|$. 11
1.5 The DAG structure for overlapping group LASSO. 16

2.1 Composition of (linear) projections fails to be a proximal map. 26
2.2 Characterization of the "roundness" of the Hilbertian ball. 33
2.3 The proximal map of the Berhu regularizer. 34
2.4 Tree-structured groups are simply rooted subtrees in a rooted tree. 35
2.5 Illustration of the main idea presented in the proof of Theorem 2.6. 41
2.6 An admissible function f that is *not* increasing *w.r.t.* the norm. 43

3.1 Comparison of the Moreau envelop and the proximal average. 55
3.2 Objective value vs. iteration on overlapping group lasso. 63
3.3 Objective value vs. iteration on graph-guided fused lasso. 63

4.1 The duality gap. 67
4.2 The gauge through the Minkowski functional. 76
4.3 The idea of convexifying dictionary learning. 85
4.4 Training and test performance on MovieLens100k. 91
4.5 Training and test performance on MovieLens1M. 91
4.6 Training and test performance on MovieLens10M. 91
4.7 Multi-class classification on the synthetic dataset. 92
4.8 Multi-task learning on the school dataset. 92

List of Symbols

$\langle \cdot, \cdot \rangle$	inner product (and more generally, a dual pairing)	1
sign	the sign function	1
$(\cdot)_+$	positive part	2
$\|\cdot\|_2$	ℓ_2 norm	2
$(\cdot)^\top$	matrix transpose	2
$\|\cdot\|_0$	the number of nonzero entries	3
$\|\cdot\|_1$	ℓ_1 norm	3
$\|\cdot\|_\infty$	ℓ_∞ norm	4
$\mathcal{H}$	Hilbert space, usually the underlying domain	6
$\|\cdot\|_{\mathcal{H}}$	the Hilbertian norm	6
$\|\cdot\|_\circ, \|\cdot\|^\circ$	polar (dual norm) of $\|\cdot\|$	6
ι_C	$\{0, \infty\}$-valued indicator function of the set C	7
dom	effective domain	7
Γ_0	the set of all closed proper convex functions	7
$\partial F(\mathbf{w})$	the generalized gradient of the locally Lipschitz function F at point $\mathbf{w}$; reduces to the subdifferential if F is convex	7
$\nabla F(\mathbf{w})$	the gradient of F at point $\mathbf{w}$	7
argmin	the set of minimizers	7
η_t	the step size at time step t, nonnegative	7
M	the Lipschitz constant (of the function) or the bound on the subdifferential	8
L	the Lipschitz constant of the gradient *w.r.t.* some norm	9
Id	the identity map	9
P_f^η	the proximal map of the function f	10
M_f^η	the Moreau envelop of the function f	10
F^*	Fenchel conjugate of F	14
$\|\cdot\|_p$	the ℓ_p norm	16
$\|\cdot\|_{\mathsf{TV}}$	the total variation seminorm	17
$\|\cdot\|_{\mathrm{tr}}$	trace norm, sum of singular values	19
$\|\cdot\|_F$	Frobenius norm	19

$f \circ g$	the composition of f and g (applying g first)	25
$\dim$	dimension of the object	29
$\mathbf{x} \perp \mathbf{y}$	perpendicular: $\langle \mathbf{x}, \mathbf{y} \rangle = 0$	31
κ	a gauge (a positively homogenenous convex function)	31
q	the quadratic function $\frac{1}{2} \|\cdot\|_{\mathcal{H}}^2$	32
2^I	the power set of the set I	35
SC_η	the class of η-strongly convex functions	52
SS_η	the class of finite-valued functions with η-Lipschitz continuous gradient	52
$\mathsf{A}^\eta_{\mathbf{f},\boldsymbol{\alpha}}, \mathsf{A}^\eta$	the proximal average of $\mathbf{f} = (f_1, \ldots, f_K)$ under the weight $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$.	53
$\mathrm{Range}(f)$	the range (image) of the map f	66
$\mathsf{G}(\mathbf{w})$	the duality gap at point $\mathbf{w}$	66
$\tilde{\mathsf{G}}_t$	the minimal duality gap	73
κ°	the polar of the gauge κ	76
$\mathcal{A}$	the atomic set	76
conv	the (closed) convex hull	76
$\|\cdot\|_c$	the column norm	83
$\|\cdot\|_r$	the row norm	84
$U_{:i}$	i-th column of the matrix U	84
$V_{i:}$	i-th row of the matrix V	84

List of Abbreviations

SVM	Support Vector Machines	2
LASSO	Least Absolute Shrinkage and Selection Operator	3
LP	Linear Programming	5
w.r.t.	with respect to	8
PG	Proximal Gradient	9
APG	Accelerated Proximal Gradient	11
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm	11
PSG	Proximal Subgradient	13
RDA	Regularized Dual Averaging	13
GCG	Generalized Conditional Gradient	14
DAG	Directed Acyclic Graph	17
SDP	Semedefinite Programming	19
SVD	Singular Value Decomposition	19
s.p.d.	self-prox-decomposable	27
p.h.	positively homogeneous	30
w.l.o.g.	without loss of generality	37
l.s.c.	lower semicontinuous	40
u.s.c.	upper semicontinuous	42
S-APG	Smoothed Accelerated Proximal Gradient	47
PA-APG	Proximal Average based Accelerated Proximal Gradient	49
PA-PG	Proximal Average based Proximal Gradient	49

Chapter 1

Introduction

Many problems in machine learning fall into the regularized empirical risk minimization framework:

$$\inf_{\mathbf{w}\in C} F(\mathbf{w}), \quad \text{where} \quad F(\mathbf{w}) = \ell(\mathbf{w}) + f(\mathbf{w}), \tag{1.1}$$

with $C \subseteq \mathcal{H}$ the parameter space (say, $\mathcal{H} = \mathbb{R}^m$, the m-dimensional Euclidean space), ℓ the loss function that encodes our preference over different parameters $\mathbf{w}$, and f the regularizer that induces some desired structure on the parameters. Usually there is some trade-off parameter $\lambda \geq 0$ that balances the two different goals; here we have chosen to absorb this constant in the regularizer f.

Due to its apparent importance, problem (1.1) has been extensively studied and a lot of algorithms have been proposed. In this chapter, we first present some motivating examples that fall into the framework of (1.1)—demonstrating its ubiquity. Then we review four popular algorithms for solving (1.1), where a common key component is the utilization of the proximal map of the regularizer. Next, through a sequence of structured sparse regularizers, we show that this proximal map, unfortunately, is not always easily computable. Thus the main goal of this thesis is to develop more efficient algorithms for computing the proximal map, through either a detailed analysis of its properties, or making certain linear approximation of it, or even a completely different algorithm that bypasses it. We end this chapter with a summary of the main contributions made in this thesis.

This chapter does not contain any new result.

1.1 Examples

We collect here some important machine learning examples that fall into the regularized empirical risk minimization framework (1.1). These examples are meant to be motivating but not exhaustive.

Example 1.1 (Binary Classification, Devroye et al. 1996)**.** *In this example, we are given training data* $(\mathbf{x}_i, y_i), i = 1, \ldots, n$*, where the covariate* $\mathbf{x}_i \in \mathbb{R}^m$*, and the label* $y_i \in \{-1, 1\}$*. We want to learn a linear classifier* $h_{\mathbf{z},b}(\mathbf{x}) = \operatorname{sign}(\langle \mathbf{x}, \mathbf{z}\rangle + b)$*, where throughout* $\langle \cdot, \cdot \rangle$ *denotes the inner product (with the underlying space clear from context),* $b \in \mathbb{R}$ *is a bias term, and* sign *is the sign function,* i.e.*,* $\operatorname{sign}(z) = 1$ *if* $z \geq 0$ *and* $\operatorname{sign}(z) = -1$ *otherwise. We minimize the empirical risk under the*

zero-one loss:

$$\min_{(\mathbf{z},b)\in\mathbb{R}^{m+1}} \frac{1}{n}\sum_{i=1}^{n} \frac{1-\operatorname{sign}[y_i(\langle \mathbf{x}_i,\mathbf{z}\rangle + b)]}{2}.$$

Putting into the framework (1.1)*, we identify* $\mathbf{w}=(\mathbf{z},b)$*,* $C=\mathbb{R}^{m+1}$*, and* $f\equiv 0$*.*

Example 1.2 (Support Vector Machines (SVM), Cortes and Vapnik 1995)**.** *Similar as the previous example, except that we minimize under the hinge loss* $\Delta(\hat{y},y)=(1-y\hat{y})_+$*, where* $z_+:=\max\{z,0\}$ *denotes the positive part:*

$$\min_{(\mathbf{z},b)\in\mathbb{R}^{m+1}} \frac{1}{n}\sum_{i=1}^{n}[1-y_i(\langle \mathbf{x}_i,\mathbf{z}\rangle + b)]_+ + \lambda \|\mathbf{z}\|_2 . \tag{1.2}$$

We have also added the ℓ_2 *norm*[1] *regularization* $\lambda\|\mathbf{z}\|_2$*, which helps controlling the model complexity and induces the representer theorem (Steinwart and Christmann 2008). Notice that the bias term* b *is not regularized. Clearly, by identifying* $\mathbf{w}=(\mathbf{z},b)$*,* $C=\mathbb{R}^{m+1}$ *and* $f(\mathbf{w})=\lambda\|\mathbf{z}\|_2$*, we fall again to the framework* (1.1)*.*

Compared with Example 1.1, the SVM formulation (1.2) is usually preferred since both the loss term (average of hinge losses) and the regularizer are convex functions (*cf.* Definition 1.2 below), hence an approximate minimizer can be found in polynomial time (Nesterov 2003). However, due to the non-differentiability of the hinge loss and also the regularizer, a naive implementation would not scale to large datasets. Of course, one can apply the squaring trick here, that is, consider the squared hinge loss $(1-y\hat{y})_+^2$, which is smooth. Similarly, we can use the squared ℓ_2 norm $\|\mathbf{z}\|_2^2$, which amounts to an appropriate change of the constant λ. It is also possible to use, for instance, the logistic loss

$$\Delta(\hat{y},y)=\log_2(1+\exp(-y\hat{y})), \tag{1.3}$$

which is again smooth. Both the hinge loss and the logistic loss are convex upper bounds of the zero-one loss, as shown in Figure 1.1. However, as can be imagined, there will be some statistical consequences when we change the loss term (Steinwart 2007).

Another example of (1.1) comes from high dimensional statistics.

Example 1.3 (Subset Selection, Miller (2002))**.** *As before, given training data* $(\mathbf{x}_i,y_i)$*,* $i=1,\ldots,n$*, where the covariate* $\mathbf{x}_i\in\mathbb{R}^m$ *and the response* $y_i\in\mathbb{R}$*, we want to fit the data with a linear hyperplane,* i.e.*, finding some* $\mathbf{w}\in\mathbb{R}^m$ *such that* $X\mathbf{w}\approx\mathbf{y}$*, where*[2] $X=[\mathbf{x}_1,\ldots,\mathbf{x}_n]^\top\in\mathbb{R}^{n\times m}$ *and* $\mathbf{y}=[y_1,\ldots,y_n]^\top\in\mathbb{R}^n$*. In high dimensional statistics, we are interested in the case* $m\gg n$*,* i.e.*, the number of features is much larger than the number of training samples, leading to an ill-posed problem. Inevitably, we need to pose some structural assumption on the model parameter* $\mathbf{w}$*, so that the problem is at least unambiguous. One prominent such prior is sparsity: Among all*

[1] Recall that the ℓ_2 norm (*a.k.a.* the Euclidean norm) $\|\cdot\|_2$ is defined as $\|\mathbf{w}\|_2=\sqrt{\sum_i w_i^2}$.

[2] Throughout the thesis we use $A^\top$ to denote the transpose of the real matrix A.

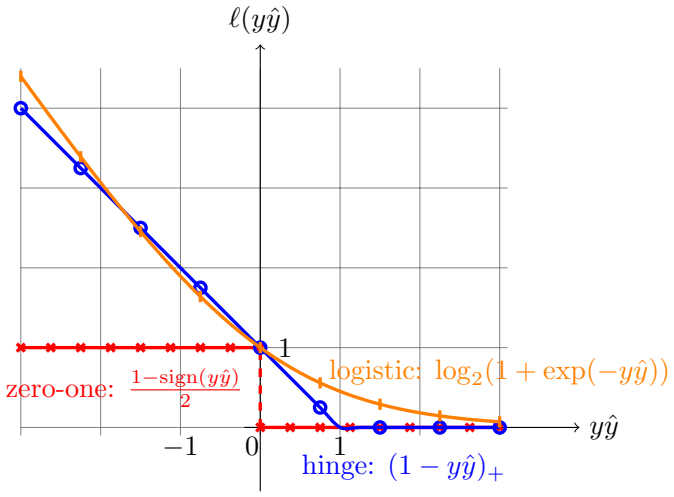


Figure 1.1: The zero-one, hinge, and logistic loss, as functions of $y\hat{y}$.

parameters that approximate the training data well, we look for the one that has the smallest number of nonzero entries:

$$\min_{\mathbf{w}\in\mathbb{R}^m} \tfrac{1}{2}\|X\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_0\,, \tag{1.4}$$

where $\|\mathbf{w}\|_0$ denotes the number of nonzero entries in $\mathbf{w}$. We have no difficulty in casting the above minimization into the framework (1.1).

Enforcing sparsity leads to multiple benefits, such as: it naturally restricts the model complexity hence avoids overfitting; it leads to more interpretable results; and it helps saving storage space; etc. However, due to the nonconvexity of the regularizer $\|\cdot\|_0$, directly solving (1.4) is hard (Natarajan 1995). Therefore, in practice one usually turns to greedy algorithms or convex relaxations.

Example 1.4 (Least Absolute Shrinkage and Selection Operator (LASSO), Tibshirani 1996)**.** *The setting is similar to Example 1.3. Inspired by the two-stage method known as nonnegative garrote (Breiman 1995), Tibshirani (1996) proposed to replace the nonconvex regularizer $\|\cdot\|_0$ with the ℓ_1 norm*[3] *constraint :*

$$\min_{\mathbf{w}\in\mathbb{R}^m} \tfrac{1}{2}\|X\mathbf{w}-\mathbf{y}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \le \zeta, \tag{1.5}$$

which, is known to be equivalent to

$$\min_{\mathbf{w}\in\mathbb{R}^m} \tfrac{1}{2}\|X\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_1\,, \tag{1.6}$$

up to an appropriate change of the constants ζ and λ. Clearly, both (1.5) *and* (1.6) *are instances of the framework* (1.1)*. Note that the ℓ_1 norm (in fact, any norm) is convex but nondifferentiable*

[3]Defined as $\|\mathbf{w}\|_1 = \sum_i |w_i|$.

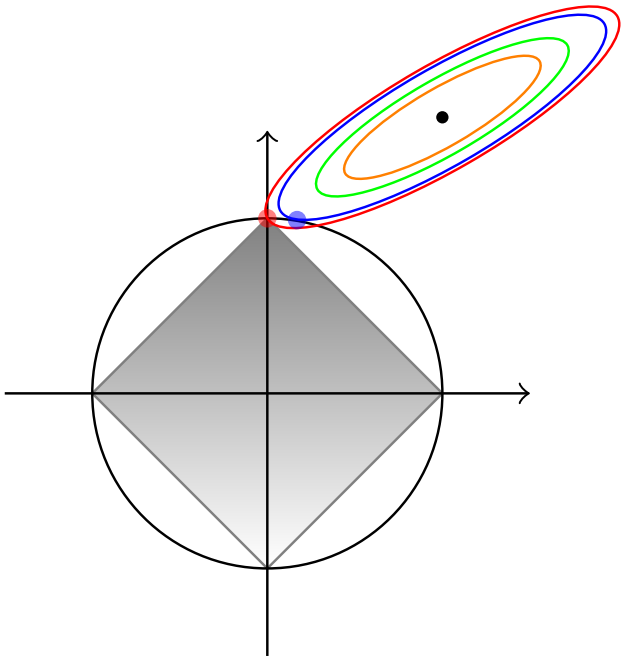


Figure 1.2: The "magic" of the ℓ_1 norm. The shaded area is the unit ball of the ℓ_1 norm while the black circle denotes the unit sphere of the ℓ_2 norm. The colored ellipses represent the sublevel sets of the loss ℓ. The "touch" point of the smallest ellipse to the unit ball is the minimizer of (1.5) (with $\zeta = 1$). In this example the ℓ_1 norm leads to a sparse solution (the red point) while the ℓ_2 norm does not (the blue point).

at the origin, and the squaring trick won't help here. Through a sequence of careful experiments, Tibshirani (1996) showed that the Lasso is capable of doing variable selection[4] *in linear regression, and through trading bias with variance it often improves the prediction accuracy.*

A heuristic argument for the effectiveness of the ℓ_1 norm relaxation is that its unit ball is "pointy". Thus it is very likely that the (sub)level sets of the loss will hit some pointy corner—a sparse minimizer in this case. In contrast, the unit ball of the ℓ_2 norm is round hence it is equally possible to hit any point. See Figure 1.2 for a vivid demonstration. As it turns out, the ℓ_1 norm is the tightest convex lower bound of the function $\|\cdot\|_0$, when restricted to the unit ball of the ℓ_∞ norm[5], see Figure 1.3. More generally, in Appendix A we show how to derive the "tightest" convex relaxation of computationally "hard" regularizers.

A very related example comes from signal processing:

Example 1.5 (Basis Pursuit, Chen et al. (2001))**.** *The motivation here is to decompose a signal* $\mathbf{s} \in \mathbb{R}^n$ *into a linear combination of "atoms"* ϕ_i *coming from a given dictionary* $\Phi = [\phi_1, \ldots, \phi_m] \in \mathbb{R}^{n \times m}$*, for instance, the canonical basis in* $\mathbb{R}^n$ *or the Fourier basis or some wavelet basis. Impor-*

[4] More precisely, as observed in Tibshirani (1996) and later thoroughly studied in Bühlmann and van de Geer (2011), Lasso does variable screening instead of selection, *i.e.*, Lasso always selects the relevant variables but potentially may include some others.

[5] Defined as $\|\mathbf{w}\|_\infty = \max_i |w_i|$.

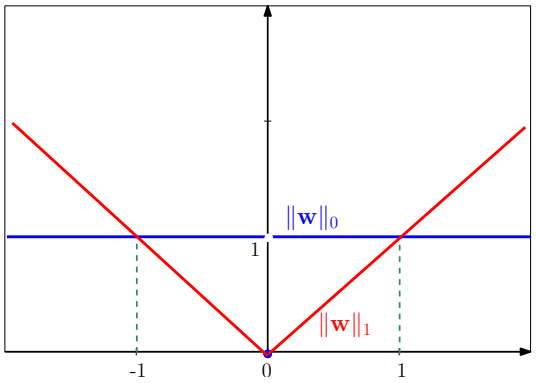


Figure 1.3: The ℓ_1 norm relaxation of $\|\cdot\|_0$.

tantly, for a variety of reasons, such as robustness, sparsity, modeling capacity, etc.*, it is often desirable to have redundancy in an ideal dictionary, that is, an overcomplete one with $m > n$ (Mallat 2009). However, mathematically, an overcomplete dictionary makes the decomposition non-unique, thus basis pursuit aims at finding the sparsest one by solving:*

$$\min_{\mathbf{w}\in\mathbb{R}^m} \|\mathbf{w}\|_1, \quad \text{s.t.} \quad \Phi\mathbf{w} = \mathbf{s}. \tag{1.7}$$

Again, the ℓ_1 norm is employed as a convex relaxation of the cardinality function $\|\cdot\|_0$. As noted by Chen et al. (2001), (1.7) is essentially an instance of linear programming (LP), and there is always a solution with at most n nonzeros. On the downside, an overcomplete dictionary increases the computational burden. However, many known dictionaries, such as the Fourier basis and some wavelet basis, admit fast matrix-vector multiplications, enabling Chen et al. (2001) to solve (1.7) on dictionaries that are thousands by tens of thousands. A crucial observation made in Chen et al. (2001) and further pursued in Donoho and Huo (2001) is that (1.7), albeit a convex relaxation*, often "magically" yields exact recovery when the signal is truly formed in a sparse way. A lot of recent work in the newly formed compressed sensing field has been devoted to explaining this phenomenon, with a shift to random dictionaries, see the recently published book of Foucart and Rauhut (2013) and the many references therein. Taking a further step, Olshausen and Field (1996) considered learning the dictionary simultaneously with the decomposition.*

By interpreting the constraint as the loss function ℓ, (1.7) falls into the framework (1.1). More generally, to accommodate noise, one turns instead to

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1, \quad \text{s.t.} \quad \|\Phi\mathbf{w} - \mathbf{s}\|_2 \le \delta,$$

for some $\delta \ge 0$, or its Lagrangian counterpart

$$\min_{\mathbf{w}} \tfrac{1}{2}\|\Phi\mathbf{w} - \mathbf{s}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

for some $\lambda \ge 0$. Note the similarity with Lasso in (1.6).

Yet another motivation for adopting the ℓ_1 norm comes from *robustness*. It is long known that the ℓ_1 loss is less affected by extremal erroneous observations than the ℓ_2 loss[6]. However, due to its closed-form solution, least squares has been dominating in many scientific areas (perhaps even today). A modern advocate of the ℓ_1 loss, more generally, the quantile regression method, is the work of Portnoy and Koenker (1997), who skillfully combined the then-groundbreaking interior-point algorithm (Nesterov and Nemiroviskii 1994), probabilistic analysis (instead of the more usual worst-case analysis) and an active set technique to demonstrate that ℓ_1 regression can be made computationally even more efficient than least squares, on problem sizes 20,000–120,000.

While it is certainly interesting to note that the ℓ_1 norm idea flourished almost at the same time in various fields, it is not entirely by chance: the emergence of interior-point algorithms made the computation affordable[7]. However, as reflected by Tibshirani (2011), the computational advance was still at shortage and called for further research.

Through the above examples, we have demonstrated the ubiquity of the framework (1.1) in machine learning (and related fields). Consequently, tremendous amount of effort has been devoted to designing better and faster algorithms. Partly reflecting this is the recent monograph edited by Sra et al. (2012). Also, in the regime of *big data*, meaning huge amounts of data which can be of ultrahigh dimension, first order optimization methods are much preferred to interior-point algorithms (which are second order), thanks to their low per-iteration complexity. Since the main contributions of this thesis are about the former, let us first review four important algorithms in that class.

1.2 Subgradient Descent

The subgradient descent algorithm (Shor 1985) is a generic procedure for minimizing convex functions, smooth or not. Let us start with recalling some definitions that will be used throughout. Let our domain be a Hilbert space[8] $\mathcal{H}$ equipped with the inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\|\cdot\|_{\mathcal{H}}$. Many results in this chapter can be generalized to a non-Hilbertian setting, such as a Banach space. We will use the norm $\|\cdot\|_{\mathcal{H}}$ to signify the Hilbertian setting and an abstract norm $\|\cdot\|$ to indicate the general setting. Note that the polar (dual norm) is defined as $\|\mathbf{g}\|_\circ = \sup_{\|\mathbf{w}\| \leq 1} \langle \mathbf{w}, \mathbf{g} \rangle$. The resulting Cauchy-Schwarz inequality $\langle \mathbf{w}, \mathbf{g} \rangle \leq \|\mathbf{w}\| \, \|\mathbf{g}\|_\circ$ is useful. The polar of $\|\cdot\|_{\mathcal{H}}$ is itself.

Definition 1.1 (Convex Set)**.** *A set $C \subseteq \mathcal{H}$ is called convex iff for all $\mathbf{x}, \mathbf{y} \in C$ and $\lambda \in]0, 1[$, we have $\lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in C$.*

Definition 1.2 (Convex Function)**.** *A function $F : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is called convex iff for all*

[6] Perhaps a bit surprisingly, regression based on the ℓ_1 loss can be traced back to Boscovich and Simpson in 1760 while least squares was popularized by Gauss "only" in 1821. Of course, one cannot take this too seriously: As pointed out by Stigler (1984), Simpson himself had already considered least squares as early as 1756. We can never be sure how far the origins can be traced to.

[7] And of course the Internet has made the dissemination of knowledge easier, cheaper and quicker.

[8] At times we will break this rule; whether or not $\mathcal{H}$ is truly Hilbertian depends on its norm, $\|\cdot\|_{\mathcal{H}}$ or $\|\cdot\|$.

$\mathbf{x}, \mathbf{y} \in \mathcal{H}$ *and* $\lambda \in]0,1[$*, we have*

$$F(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda F(\mathbf{x}) + (1-\lambda) F(\mathbf{y}). \tag{1.8}$$

The convention to let F take the value ∞ for points outside of its domain proves to be convenient. For instance, we can identify a convex set C with the indicator function

$$\iota_C(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \in C \\ \infty, & \text{otherwise} \end{cases}. \tag{1.9}$$

Under this convention, $\operatorname{dom} F := \{\mathbf{x} \in \mathcal{H} : F(\mathbf{x}) < \infty\}$ signifies the (effective) domain, which is necessarily a convex set if F is a convex function. To exclude triviality, we will only consider *proper* functions—those with nonempty domain. For regularity purpose, we assume the convex function F is *closed*, *i.e.* lower semicontinuous[9]. Collectively we use $\Gamma_0(\mathcal{H})$, or simply Γ_0 if no confusion is caused, to denote the set of all closed proper convex functions. The subdifferential of the convex function F at point $\mathbf{x}$ is defined as the set

$$\partial F(\mathbf{x}) := \{\mathbf{g} \in \mathcal{H} : F(\mathbf{y}) \geq F(\mathbf{x}) + \langle \mathbf{y} - \mathbf{x}, \mathbf{g} \rangle, \forall \mathbf{y}\}. \tag{1.10}$$

Note that the subdifferential is always a (weak*) closed convex set, possibly containing more than one element. For instance, as readily verified from the definition, the subdifferential of the absolute function $|\cdot|$ at origin is the interval $[-1, 1]$. It is also possible to have empty subdifferential at some boundary points: An example would be the subdifferential at origin of the function $-\sqrt{x}$ defined on $x \geq 0$. Notably, as long as F is continuous (and finite-valued) at $\mathbf{x}$, it can be shown that the subdifferential $\partial F(\mathbf{x})$ is nonempty (Zălinescu 2002, Theorem 2.4.12), in which case, any element in $\partial F(\mathbf{x})$ is called a subgradient of F at the point $\mathbf{x}$. It is also known that F is differentiable at $\mathbf{w}$ iff $\partial F(\mathbf{w})$ contains exactly one element, in which case we use the notation $\nabla F(\mathbf{w})$. A very useful rule, which is also easily verified from the definition of subdifferential, is that

$$\mathbf{w}^\star \in \operatorname*{argmin}_{\mathbf{w}} F(\mathbf{w}) \iff \mathbf{0} \in \partial F(\mathbf{w}^\star). \tag{1.11}$$

Subgradient descent is an extremely simple iterative algorithm. Instantiating to (1.1), each iteration amounts to[10]

$$\mathbf{w}_{t+1} = \mathsf{P}_C(\mathbf{w}_t - \eta_t \cdot \mathbf{g}_t), \tag{1.12}$$

where $\eta_t \geq 0$ is the step size, $\mathbf{g}_t \in \partial F(\mathbf{w}_t)$, assuming the latter is nonempty, and

$$\mathsf{P}_C(\mathbf{z}) := \operatorname*{argmin}_{\mathbf{w} \in C} \tfrac{1}{2} \|\mathbf{w} - \mathbf{z}\|_{\mathcal{H}}^2 \tag{1.13}$$

is the Hilbertian projection onto the set C, assumed to be closed and convex here. In the case where C is the whole space, we simply have $\mathsf{P}_C(\mathbf{z}) = \mathbf{z}$.

[9] The function $F : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is lower semicontinuous iff its sublevel set $\{\mathbf{w} \in \mathcal{H} : F(\mathbf{w}) \leq \alpha\}$ is closed for all $\alpha \in \mathbb{R}$.

[10] Throughout the thesis we use bold letters to denote vectors. Note that the i-th entry of $\mathbf{w}$ is denoted as w_i, while $\mathbf{w}_i$ is some vector that may have nothing to do with $\mathbf{w}$.

Under mild conditions on the objective function and the step size, we can supply the following convergence analysis for the subgradient algorithm:

Theorem 1.1. *Assume that the set $C \subseteq \mathcal{H}$ is closed convex and the objective $F \in \Gamma_0$ has nonempty (uniformly) bounded subdifferential*[11]*, that is, $\|\mathbf{g}\|_{\mathcal{H}} \leq M$ for all $\mathbf{g} \in \partial F(\mathbf{w}), \mathbf{w} \in C$. Start with $\mathbf{w}_0 \in C$, then after T iterations of the subgradient update (1.12), for any $\mathbf{w} \in C$,*

$$\min_{0 \leq t \leq T-1} F(\mathbf{w}_t) - F(\mathbf{w}) \leq \frac{\|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}^2 + M^2 \sum_{t=0}^{T-1} \eta_t^2}{2 \sum_{t=0}^{T-1} \eta_t}. \tag{1.14}$$

Clearly, for the right-hand side of (1.14) to converge to 0, it is both sufficient and necessary[12] to have $\sum_t \eta_t = \infty, \eta_t \to 0$. It is also possible to use a constant step size if we only desire some ϵ accuracy.

Corollary 1.1. *Fix $\epsilon > 0$, $\mathbf{w}_0 \in C$, and set $\eta_t \equiv c/M^2 \cdot \epsilon$ for some constant $c \in]0, 2[$, then under the assumptions of Theorem 1.1, after at most $T = \frac{M^2 \|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}^2}{c(2-c)} \cdot \frac{1}{\epsilon^2}$ iterations of the subgradient update (1.12), there exists some $0 \leq t \leq T-1$ such that*

$$F(\mathbf{w}_t) - F(\mathbf{w}) \leq \epsilon.$$

The same claim holds for the averaged iterate $\overline{\mathbf{w}}_T = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{w}_t$.

Surprisingly, in black-box optimization, where the only information we can obtain is the function value and an arbitrary subgradient at any queried point, the $O(1/\epsilon^2)$ complexity bound in Corollary 1.1 cannot be improved (Nesterov 2003, Theorem 3.2.1), thereby justifying the optimality of the subgradient method for *generic* nonsmooth convex optimization. On the other hand, the subgradient algorithm completely ignores the composite structure in (1.1), and we will see in the next section that by exploiting this structure (as opposed to black-box optimization), we can improve the rate significantly.

1.3 Proximal Gradient

In this section we consider another first order algorithm that significantly improves the *optimal* $O(1/\epsilon^2)$ complexity of subgradient descent. Of course, such is possible only under additional assumptions. Specifically, we need

Assumption 1.1. *The objective F is in the composite form (1.1),* i.e.*, $F = \ell + f$ for some (closed, proper) convex functions ℓ and f .*

[11] For finite-valued F, this condition is equivalent to F being M-Lipschitz continuous (*w.r.t.* the norm $\|\cdot\|_{\mathcal{H}}$), that is, $|F(\mathbf{x}) - F(\mathbf{y})| \leq M \cdot \|\mathbf{x} - \mathbf{y}\|_{\mathcal{H}}$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{H}$.

[12] Necessity: Vanishing of the first term requires $\sum_t \eta_t = \infty$ while vanishing of the second term, using the Cauchy-Schwarz inequality, implies $\frac{1}{T} \sum_{t=0}^{T-1} \eta_t \to 0$, which is equivalent as $\eta_t \to 0$ (since $\eta_t \geq 0$). Sufficiency: For t sufficiently large $\eta_t^2 \leq \varepsilon \eta_t$ due to $\eta_t \to 0$.

Assumption 1.2. *The component ℓ is differentiable (on the interior of $\operatorname{dom}\ell$), and there exists some finite constant $L \geq 0$ such that the gradient $\nabla\ell$ satisfies the following inequality for all $\mathbf{w}, \mathbf{w}' \in \operatorname{dom}\ell$ (*w.r.t. *some norm $\|\cdot\|$):*

$$\ell(\mathbf{w}) \leq \ell(\mathbf{w}') + \langle \mathbf{w} - \mathbf{w}', \nabla\ell(\mathbf{w}') \rangle + \tfrac{L}{2} \|\mathbf{w} - \mathbf{w}'\|^2 . \tag{1.15}$$

The inequality (1.15) simply means that the function ℓ can be upper bounded by a quadratic. Clearly if (1.15) holds for some L, then it also holds for all $\tilde{L} \geq L$. Moreover, it is known that (1.15) holds when the gradient $\nabla\ell$ is L-Lipschitz continuous (*w.r.t.* the dual norm $\|\cdot\|_\circ$)[13], see *e.g.* Zălinescu (2002, Corollary 3.5.7).[14] Another convenient rule to check (1.15) for twice differentiable ℓ, in the Hilbertian setting, is that the eigenvalues of its Hessian are upper bounded by L (Nesterov 2003, Theorem 2.1.6). Both the least squares loss (in Example 1.4) and the logistic loss (1.3) satisfy Assumption 1.2.

The proximal gradient (PG) algorithm, first proposed by Fukushima and Mine (1981) as a linearization (and also generalization if we let $\ell \equiv 0$) of the proximal point algorithm (Martinet 1970; Rockafellar 1976) in the Hilbertian setting, is also iterative. We will motivate PG from an operator splitting point of view as follows. First recall the optimality condition (1.11) for (1.1), under Assumption 1.1 and Assumption 1.2:

$$0 \in \nabla\ell(\mathbf{w}^\star) + \partial f(\mathbf{w}^\star), \tag{1.16}$$

where $\mathbf{w}^\star$ is some assumed minimizer. That is, we are looking for an "annihilator" of the *sum* of two operators: $\nabla\ell$ and ∂f. It follows from simple algebra that $\mathbf{w}^\star - \eta \cdot \nabla\ell(\mathbf{w}^\star) \in (\mathsf{Id} + \eta \cdot \partial f)(\mathbf{w}^\star)$, where Id denotes the identity map. Thus[15]

$$\mathbf{w}^\star = (\mathsf{Id} + \eta \cdot \partial f)^{-1}(\mathbf{w}^\star - \eta \cdot \nabla\ell(\mathbf{w}^\star)). \tag{1.17}$$

So we have arrived at a fixed-point equation, which also splits the two operators into two consecutive steps. Quite naturally, with some initial point, we can repeatedly apply the fixed-point equation. Hopefully the generated sequence will converge to a minimizer.

PG exactly realizes the above fixed-point iteration. It simply aims at minimizing the quadratic upper bound in (1.15), with the regularizer untouched:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \ell(\mathbf{w}_t) + \langle \mathbf{w} - \mathbf{w}_t, \nabla\ell(\mathbf{w}_t) \rangle + \tfrac{1}{2\eta_t} \|\mathbf{w} - \mathbf{w}_t\|_{\mathcal{H}}^2 + f(\mathbf{w}). \tag{1.18}$$

For clarity, let us decompose the above into two steps:

$$\mathbf{z}_t = \mathbf{w}_t - \eta_t \nabla\ell(\mathbf{w}_t), \tag{1.19}$$

[13] Meaning that $\|\nabla\ell(\mathbf{x}) - \nabla\ell(\mathbf{y})\| \leq L \cdot \|\mathbf{x} - \mathbf{y}\|_\circ$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{H}$.

[14] Many references including Zălinescu (2002) require ℓ to be finite-valued, but the proof trivially extends to infinite-valued ℓ. The converse, that is, (1.15) implies the Lipschitz continuity of $\nabla\ell$ in the case where ℓ is convex, seems to require ℓ to be finite-valued.

[15] It is not entirely obvious why we get equality here. For the sake of motivation, let us not dwell on rigor.

$$\mathbf{w}_{t+1} = \mathsf{P}_f^{\eta_t}(\mathbf{z}_t) := \operatorname*{argmin}_{\mathbf{w}} \tfrac{1}{2\eta_t} \|\mathbf{w} - \mathbf{z}_t\|_{\mathcal{H}}^2 + f(\mathbf{w}). \tag{1.20}$$

The equivalence to the fixed-point equation (1.17) is verified by applying the optimality condition (1.11) to (1.20). Note that by possibly redefining f we can assume that the constraint set C is the whole space. The first step is a simple gradient update, taking only the smooth part ℓ into account; the second step is simply the proximal map (Moreau 1965) of the other part f. The proximal map has many interesting properties, some of which will be thoroughly discussed later. For now, it is enough to notice that when $f = \iota_C$, the $\{0, \infty\}$-valued indicator function of the closed convex set C, the proximal map reduces exactly to the Hilbertian projection onto C, in which case we recover the projected gradient algorithm of Goldstein (1964).

The proximal gradient algorithm has been extensively studied in recent years, see *e.g.* (Beck and Teboulle 2009; Combettes and Wajs 2005; Nesterov 2013; Tseng 2008, 2010) and the many references therein. In particular, we have the following convergence result:

Theorem 1.2. *Let Assumption 1.1, Assumption 1.2 hold and assume further that* $\operatorname{dom} \ell \supseteq \operatorname{dom} f$. *Start with* $\mathbf{w}_0 \in \operatorname{dom} f$ *and choose some constant step size* $\eta_t \equiv \eta \in]0, 1/L[$. *Then for any* $\mathbf{w}$,

$$F(\mathbf{w}_t) \leq F(\mathbf{w}) + \frac{\|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}^2}{2\eta t}.$$

Needless to say that the same rate holds in the special case $f = \iota_C$ for some closed and convex set C, corresponding to the projected gradient algorithm of Goldstein (1964). Evidently, PG is significantly faster: $O(1/\epsilon)$ versus the $O(1/\epsilon^2)$ complexity of the subgradient descent, *cf.* Corollary 1.1, provided that we can very quickly compute the proximal map (1.20) in each iteration. Such is the case when the regularizer f is "simple"—a point can be easily made by revisiting the LASSO example; for more examples, see Combettes and Pesquet (2011), Bach et al. (2011, §3.3), Parikh and Boyd (2013, §6).

Example 1.4 (continuing from p. 3)**.** *Clearly, both Assumption 1.1 and Assumption 1.2 are satisfied in this example. The first step of PG is easy:*

$$\mathbf{z}_t = \mathbf{w}_t - \eta X^\top (X\mathbf{w}_t - \mathbf{y}).$$

The second step, known as the soft-thresholding or shrinkage operator, can be computed in closed-form:

$$[\mathsf{P}_{\lambda f}^{\eta}(\mathbf{z})]_i = [\mathsf{P}_f^{\lambda\eta}(\mathbf{z})]_i = z_i\big(1 - \lambda\eta/|z_i|\big)_+. \tag{1.21}$$

See Figure 1.4 for a one dimensional example (with the understanding $\eta\lambda = \mu$*). There,* M_f^μ*, the Moreau envelop, is the minimum value on the right-hand side of* (1.20)*. In this case,*

$$\mathsf{M}_f^\mu(z) = \begin{cases} \frac{1}{2\mu} z^2, & \text{if } |z| \leq \mu \\ |z| - \frac{\mu}{2}, & \text{otherwise} \end{cases}$$

coincides with the so-called Huber's loss in robust statistics (Huber 1964).

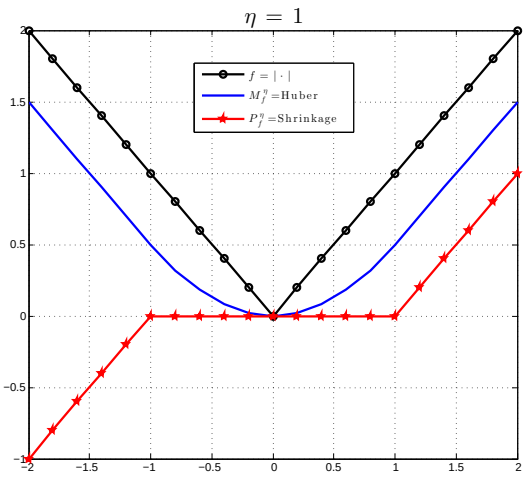


Figure 1.4: The Moreau envelop and the proximal map of the absolute function $|\cdot|$.

PG, in this specific form with ℓ_1 norm regularization, appeared first in Starck et al. (2003) and Figueiredo and Nowak (2003), although its formal convergence was established later in Daubechies et al. (2004). Surprisingly, Daubechies et al. (2004) actually proved strong[16] *convergence while it was known that PG, in general, may fail to converge strongly (Güler 1991). Further improvement can be found in Combettes and Pesquet (2007). In the finite dimensional setting, Tseng (2010) proved that in fact PG (on this example) eventually converges at a linear rate after an unspecified number of steps.*

Example 1.4 also reveals a nice property about the proximal gradient algorithm. First recall that the regularizer f in machine learning is usually employed for realizing useful structural priors on the parameters. As we will show later, there is a 1-1 correspondence between the regularizer f and its proximal map P_f^η. Therefore the nice properties of the regularizer f may be reflected in its proximal map, thus easily exploited by PG. Back to Example 1.4, the ℓ_1 norm regularizer is utilized to promote sparsity; indeed its proximal map shrinks small components to zero. This feature is in sharp contrast with the *generic* subgradient method which does not produce sparse intermediate iterates, thus partly explains why PG, besides its fast convergence, is so popular in machine learning applications.

Very surprisingly, the rate of PG is not optimal and a slight modification of the algorithm could further improve it to $O(1/\sqrt{\epsilon})$. The first variant along this direction is due to Beck and Teboulle (2009) although the main idea traces back to Nesterov (1983). Following Tseng (2008), we call these fast variants accelerated proximal gradient (APG). In particular, the algorithm of Beck and Teboulle (2009), widely known as FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), is given in Algorithm 1, and we summarize its convergence property in the next theorem.

[16] By strong convergence we mean the usual convergence under the norm, which is to be contrasted with the "weak" convergence induced by all continuous linear functionals. The two are the same in a finite dimensional space but differ fundamentally in infinite dimensional spaces.

Algorithm 1 FISTA (Beck and Teboulle 2009).

1: Initialize: $\mathbf{w}_0 = \mathbf{u}_1 \in \operatorname{dom} F$, $\eta = 1/L$, $\gamma_1 = 1$.
2: **for** $t = 1, 2, \ldots$ **do**
3: $\quad \mathbf{z}_t = \mathbf{u}_t - \eta \nabla \ell(\mathbf{u}_t)$
4: $\quad \mathbf{w}_t = \mathsf{P}_f^{\eta}(\mathbf{z}_t)$
5: $\quad \gamma_{t+1} = \frac{1+\sqrt{1+4\gamma_t^2}}{2}$
6: $\quad \mathbf{u}_{t+1} = \mathbf{w}_t + \frac{\gamma_t - 1}{\gamma_{t+1}}(\mathbf{w}_t - \mathbf{w}_{t-1})$
7: **end for**

Theorem 1.3. *Under Assumption 1.1, Assumption 1.2, and assume* $\operatorname{dom} \ell \supseteq \{2\mathbf{w} - \mathbf{w}' : \mathbf{w}, \mathbf{w}' \in \operatorname{dom} f\}$. *Start with* $\mathbf{w}_0 \in \operatorname{dom} F$ *and let* $\mathbf{w}$ *be arbitrary. The iterates produced by Algorithm 1 satisfy*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}) + \frac{2L \left\| \mathbf{w}_0 - \mathbf{w} \right\|_{\mathcal{H}}^2}{(t+1)^2}.$$

It is quite remarkable that a simple extrapolation of the iterates $\mathbf{w}_t$ (line 6 in Algorithm 1) immediately boosts the convergence rate from $O(1/\epsilon)$ to $O(1/\sqrt{\epsilon})$, although a clear intuitive explanation is not available. We note that Algorithm 1 requires the smooth part ℓ to be defined on $\{2\mathbf{w} - \mathbf{w}' : \mathbf{w}, \mathbf{w}' \in \operatorname{dom} f\}$ as the extrapolation (line 6 in Algorithm 1) may go outside of $\operatorname{dom} f$ (whereas line 4 in Algorithm 1 guarantees $\mathbf{w}_t \in \operatorname{dom} f$). There are other variants that avoid this issue, see Tseng (2008) for more discussions.

When the Lipschitz constant L is not known in advance, we can employ an adaptive backtracking strategy, see Beck and Teboulle (2009); Nesterov (2013) for detailed discussions. Wright et al. (2009) combined the line search procedure of Barzilai and Borwein (1988) with PG, and claimed superior performance on some signal processing experiments.

It might appear that APG should always be preferred over PG, since it enjoys faster convergence rates which are usually confirmed in practice. However, experience warns us of drawing any conclusion of this type. Indeed, the additional extrapolation step in APG makes analyzing its iterates much more difficult. In contrast, under fairly loose conditions on the step size and the finite-valued assumption on ℓ, it is easy to prove that the iterates generated by PG converges (weakly) to some minimizer (Combettes and Wajs 2005).

1.4 Proximal Subgradient

The PG algorithm in the previous section replaces the Hilbertian projection in the projected gradient algorithm of Goldstein (1964) with the more general proximal map (Moreau 1965). Straightforwardly, it is tempting to recycle the same idea on the Hilbertian projection in the projected *subgradient* algorithm that we saw in Section 1.2. The resulting algorithm simply abandons the smoothness Assumption 1.2 and uses an arbitrary subgradient in the first step of PG. Consequently, we need to adopt a diminishing step size (as the accuracy requirement ϵ goes to 0), and bear a slower rate of con-

Algorithm 2 PSG (Duchi and Singer 2009).

1: Initialize: $\mathbf{w}_0 \in \operatorname{argmin} f$, $\eta_0 \geq 0$.
2: **for** $t = 0, 1, \ldots$ **do**
3: $\quad \mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w}} \langle \mathbf{w} - \mathbf{w}_t, \partial \ell(\mathbf{w}_t) \rangle + \frac{1}{2\eta_t} \|\mathbf{w} - \mathbf{w}_t\|_{\mathcal{H}}^2 + f(\mathbf{w})$
4: $\quad\quad = \mathsf{P}_f^{\eta_t}\left(\mathbf{w}_t - \eta_t \cdot \partial \ell(\mathbf{w}_t)\right)$
5: **end for**

vergence. The resulting Algorithm 2, which we call proximal subgradient (PSG), has been explicitly studied by Duchi and Singer (2009). We record their result below.

Theorem 1.4. *Under Assumption 1.1 and assume* $\|\mathbf{g}\|_{\mathcal{H}} \leq M$ *for all* $\mathbf{g} \in \partial \ell(\mathbf{w}) \neq \emptyset$ *and* $\mathbf{w} \in \operatorname{dom} f$. *Use a constant step size* $\eta_t \equiv \eta$ *and start with* $\mathbf{w}_0 \in \operatorname{dom} f$. *Then after* T *iterations of the proximal subgradient algorithm, we have for any* $\mathbf{w} \in \operatorname{dom} F$,

$$\sum_{t=0}^{T-1} [F(\mathbf{w}_t) - F(\mathbf{w})] \leq \frac{\|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}^2 + M^2 \eta^2 T}{2\eta} + f(\mathbf{w}_0) - f(\mathbf{w}_T). \tag{1.22}$$

If one is concerned with the last term $f(\mathbf{w}_0) - f(\mathbf{w}_T)$, we can remove it by starting from $\mathbf{w}_0 \in \operatorname{argmin} f$.

Dealing with a natural generalization of the projected subgradient method (*cf.* Section 1.2), we expect to obtain the same, if not worse, rate of convergence for PSG. The next corollary indeed confirms this, hence also demonstrates that the composite structure (1.1) alone can not lead to faster rates. Note that an efficient implementation of PSG hinges on our ability to quickly solve the proximal map in line 4 of Algorithm 2—a requirement usually stronger than getting an arbitrary subgradient of f. The flip side of not linearizing f in line 3 of Algorithm 2 is the possibility to explicitly leverage any special property of the proximal map of the regularizer f, such as the sparsity we saw in Example 1.4.

Corollary 1.2. *Fix* $\epsilon > 0$. *Choose* $\mathbf{w}_0 \in \operatorname{argmin} f$ *and set* $\eta \equiv c/M^2 \cdot \epsilon$ *for some constant* $c \in]0, 2[$, *then under the assumptions of Theorem 1.4, after at most* $T = \frac{M^2 \|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}^2}{c(2-c)} \cdot \frac{1}{\epsilon^2}$ *iterations, there exists some* $0 \leq t \leq T - 1$ *such that*

$$F(\mathbf{w}_t) \leq F(\mathbf{w}) + \epsilon.$$

The same claim holds for the averaged iterate $\overline{\mathbf{w}}_T = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{w}_t$.

1.5 Regularized Dual Averaging

The regularized dual averaging (RDA) of Xiao (2010), in some sense the "dual" of PSG, is a composite generalization of the dual averaging proposed by Nesterov (2009). As summarized in Algorithm 3, RDA averages the subgradients, instead of the iterates. We will motivate and present RDA in a different way than Xiao (2010) or Nesterov (2009).

We need one more definition.

Algorithm 3 RDA (Xiao 2010).

Initialize: $\mathbf{w}_0 \in \operatorname{argmin} f$, $\bar{\mathbf{g}}_0 = \mathbf{0}$, $\beta_0 > 0$, $\alpha_t \equiv 1$.
for $t = 1, 2, \dots$ **do**
  $\mathbf{g}_{t-1} \in \partial \ell(\mathbf{w}_{t-1})$
  $\mathbf{s}_t = \mathbf{s}_{t-1} + \alpha_{t-1}\mathbf{g}_{t-1}$
  choose β_t
  $\mathbf{w}_t = \operatorname{argmin}_{\mathbf{w}} \langle \mathbf{w}, \mathbf{s}_t \rangle + \beta_t \cdot \frac{1}{2} \|\mathbf{w} - \mathbf{w}_0\|_{\mathcal{H}}^2 + (\sum_{\tau=0}^{t} \alpha_\tau) \cdot f(\mathbf{w})$
end for

Definition 1.3. *For any closed, proper and convex function $F \in \Gamma_0$, its Fenchel conjugate F^* is defined as:*

$$F^*(\mathbf{g}) = \sup_{\mathbf{w}} \langle \mathbf{w}, \mathbf{g} \rangle - F(\mathbf{w}). \tag{1.23}$$

It is easily verified that again $F^* \in \Gamma_0$. Moreover, $(F^*)^* = F$. From the definition we have the inequality

$$\langle \mathbf{g}, \mathbf{w} \rangle \leq F(\mathbf{w}) + F^*(\mathbf{g}),$$

with equality iff $\mathbf{w} \in \partial F^*(\mathbf{g})$ iff $\mathbf{g} \in \partial F(\mathbf{w})$.

By the Fenchel-Rockafellar duality (Zălinescu 2002, Corollary 2.8.5), we have the relation (under mild technical conditions)

$$\begin{aligned} \inf_{\mathbf{w}} \ell(\mathbf{w}) + f(\mathbf{w}) &= \sup_{\mathbf{g}} -\ell^*(\mathbf{g}) - f^*(-\mathbf{g}) \\ &= -\inf_{\mathbf{g}} \underbrace{\ell^*(\mathbf{g})}_{\tilde{f}(\mathbf{g})} + \underbrace{\sup_{\mathbf{w}} \langle \mathbf{w}, -\mathbf{g} \rangle - f(\mathbf{w})}_{\tilde{\ell}(\mathbf{g})}. \end{aligned} \tag{1.24}$$

In other words, we have transformed the original composite problem into a similar one in the dual, with the role of ℓ and f swapped. Of course, we can now apply PSG on this dual formulation (1.24), provided that we can compute the proximal map $\mathsf{P}_{\tilde{f}}^{1/\eta_t} = \mathsf{P}_{\ell^*}^{1/\eta_t}$, which, as we will see in the next chapter, is computationally equivalent to $\mathsf{P}_{\ell}^{1/\eta_t}$. Said differently, we could have just swapped the role of ℓ and f in the original problem—disappointingly—nothing seems to have been gained by going to the dual. This is where we need a substantially new idea.

The idea is to apply a PG-like algorithm to the dual; more precisely, the generalized conditional gradient (GCG) algorithm that we will thoroughly discuss in Chapter 4. GCG, like PG, requires the loss ℓ to be smooth, *i.e.*, satisfy Assumption 1.2. Unfortunately, $\tilde{\ell}$ in (1.24) usually is not smooth, unless the regularizer f is strongly convex. Nevertheless, we can turn instead to a *smooth* approximation of $\tilde{\ell}$. Let $\mathbf{w}_0 \in \operatorname{argmin} f$, and consider

$$\tilde{\ell}_\mu(\mathbf{g}) = \sup_{\mathbf{w}} \langle \mathbf{w}, -\mathbf{g} \rangle - f(\mathbf{w}) - \tfrac{\mu}{2} \|\mathbf{w} - \mathbf{w}_0\|_{\mathcal{H}}^2, \tag{1.25}$$

which, being the Fenchel conjugate of a μ-strongly convex function, satisfies Assumption 1.2 with $L = 1/\mu$, see, *e.g.* Zălinescu (2002, Corollary 3.5.11). Observe that (1.25) is exactly the step 6 in

Algorithm 3, up to a sign and constant change. GCG proceeds by linearizing the smooth loss $\tilde{\ell}_\mu$ in each step and finds a direction

$$\operatorname*{argmin}_{\mathbf{g}} \langle \mathbf{g}, -\mathbf{w}_{t+1} \rangle + \tilde{f}(\mathbf{g}) = \operatorname*{argmax}_{\mathbf{g}} \langle \mathbf{g}, \mathbf{w}_{t+1} \rangle - \ell^*(\mathbf{g}) = \partial \ell(\mathbf{w}_{t+1}).$$

Then GCG simply takes a convex combination of the direction above and the current iterate, that is, step 4 in Algorithm 3 (better seen if we consider the average $\tilde{\mathbf{s}}_t = \mathbf{s}_t / \sum_{\tau=0}^{t} \alpha_\tau$).

To analyze the performance of RDA, we introduce the set $Q := \{\mathbf{w} : \frac{1}{2} \|\mathbf{w} - \mathbf{w}_0\|_{\mathcal{H}}^2 \leq \Delta^2\}$ for some $\Delta > 0$. Clearly Q is bounded. We will compare the iterates of RDA with any fixed point in Q—a restriction of the original problem. Obviously, as Δ becomes large, we approach the original problem. Note that Algorithm 3 does not need to know Q at all. The performance of RDA is summarized below.

Theorem 1.5. *Under Assumption 1.1 and assume that ℓ is subdifferentiable on* $\operatorname{dom} f$. *Then for any $\mathbf{w} \in \operatorname{dom} F \cap Q$, Algorithm 3 with increasing step size $(\beta_t)_{t \geq 0} \uparrow$ satisfies*

$$\sum_{t=0}^{T-1} \ell_t(\mathbf{w}_t) + f(\mathbf{w}_t) - \ell_t(\mathbf{w}) - f(\mathbf{w}) \leq \beta_T \Delta^2 + \sum_{t=0}^{T-1} \tfrac{1}{2\beta_t} \|\mathbf{g}_t\|_\circ^2 . \tag{1.26}$$

There is an apparent trade-off in the two terms on the right-hand side of (1.26), due to the non-decreasing requirement on β_t. A careful balance, *e.g.*, $\beta_t = O(\sqrt{t})$ yields an $O(1/\sqrt{t})$ convergence rate, similar to that of PSG. Moreover, when f is itself strongly convex[17], we do not need the extra smoothing in (1.25), and the faster $O(1/t)$ rate can be proven.

1.6 Structured Sparsity

The four algorithms we discussed in the previous sections are by no means exhaustive. However, a common requirement of them is the possibility of quickly solving the proximal map (1.20) in each iteration (recall that the Hilbertian projection is a special proximal map). For the $\mathfrak{l}_1$ norm which has played a vital role in sparse estimation, its proximal map indeed can be easily computed, see Example 1.4. However, we quickly lose this nice gift once we consider more refined notions of sparsity—structured sparsity (Bach et al. 2012).

Structured sparsity, generally speaking, refers to our belief that not all sparse patterns are equally desired.[18] Some particular sparse patterns may be preferred as compared to others, and sparsity may present itself in other structured forms, in addition to the number of nonzero elements. Let us consider some examples.

Example 1.6 (Non-overlapping Group Sparsity, Bakin (1999); Yuan and Lin (2006))**.** *Motivated by the multifactor analysis-of-variance problem, where each explanatory factor is naturally formed by*

[17] Namely $f - \frac{\sigma}{2} \|\cdot\|_{\mathcal{H}}^2$ is convex for some $\sigma > 0$.

[18] Which, by the way, should not be generalized to humans.

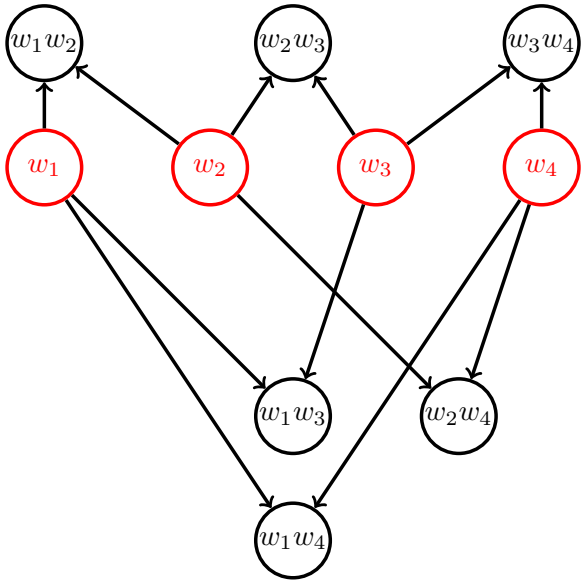


Figure 1.5: The DAG structure for overlapping group LASSO. The red nodes represent the main effect variables and the black nodes designate interactions between the main effects. The groups are all (rooted) subtrees of this DAG.

a group of derived input variables, Yuan and Lin (2006) considered the (non-overlapping) group LASSO regularizer

$$f_{\mathsf{GL}}(\mathbf{w}) = \sum_{g \in \mathcal{G}} \|\mathbf{w}_g\|, \tag{1.27}$$

where $\mathcal{G}$ is a partition *of the variables, and we use $\mathbf{w}_g$ to denote the subvector of $\mathbf{w}$ indexed by the variables belonging to group $g \in \mathcal{G}$. The norm $\|\cdot\|$ is taken to be the $\mathfrak{l}_2$ or its weighted version which takes the group size into account.*[19] *Clearly, when each group g consists of exactly one variable (or the norm $\|\cdot\|$ being $\mathfrak{l}_1$), we recover the LASSO regularizer in Example 1.4. Yuan and Lin (2006) adopted a coordinate-wise algorithm (under the least squares loss), and also a homotopy variant of the algorithm in Efron et al. (2004). Thanks to the non-overlapping property, the proximal map in this case is separable (for each group) hence easily computed. For the $\mathfrak{l}_2$ norm, we have*

$$[\mathsf{P}^{\eta}_{f_{\mathsf{GL}}}(\mathbf{w})]_i = w_i(1 - \eta/\|\mathbf{w}_g\|_2)_+, \text{ if } i \in g. \tag{1.28}$$

Example 1.7 (Hierarchical Group Sparsity, Zhao et al. (2009))**.** *Zhao et al. (2009) generalized Example 1.6 in two aspects: 1). the norm $\|\cdot\|$ can be of any $\mathfrak{l}_p$ type,*[20] *in particular $\mathfrak{l}_\infty$ that leads to some computational savings; 2). the groups $\mathcal{G}$ need not be a partition of the variables. The main motivation for 2) comes from causality: some features might be just interactions between some main*

[19] We have suppressed the possible notational dependence of the norm $\|\cdot\|$ on the group g.
[20] Recall the definition $\|\mathbf{w}\|_p = (\sum_i |w_i|^p)^{1/p}$.

effects, therefore should only be selected after the main effects are. The idea to enforce 2) is simply to allow the groups to overlap: Consider two subsets (may or may not belong to $\mathcal{G}$) of variables, say s_1, s_2. Design the groups $\mathcal{G}$ so that for every group $g \supseteq s_1$ we have $g \supseteq s_2$. In other words, s_1 is "dominated" by s_2 in the sense of group coverage. Heuristically, when s_2 is selected, all groups containing s_2 are selected, hence s_1 is selected as well due to the dominance. Zhao et al. (2009) proceeded to consider a directed acyclic graph (DAG) structure of the variables, see Figure 1.5. There, the nodes represent variables; the directions indicate causality relations; and the groups are simply all the (rooted) subtrees in this DAG.

Unfortunately, the proximal map for such an overlapping group LASSO regularizer cannot be easily computed. Very surprisingly though, when the DAG is a rooted tree, Jenatton et al. (2011) showed that the proximal map can be cleverly computed—a result motivated our development in Chapter 2. Zhao et al. (2009) developed a homotopy variant of the algorithm in Efron et al. (2004), when the group norm is ℓ_∞. The hierarchical structure is usually assumed given a priori*, and an interesting open problem is to learn this information directly from data.*

Example 1.8 (Spatial Sparsity, Kim et al. (2009); Tibshirani et al. (2005))**.** *The fused LASSO is proposed by Tibshirani et al. (2005) to respect certain ordering of the features. The regularizer is a (weighted)* sum *of the ℓ_1 norm and the total variation (semi)norm:*

$$\|\mathbf{w}\|_{\mathsf{TV}} = \sum_{i=2}^{m} |w_i - w_{i-1}|, \tag{1.29}$$

$$f_{\mathsf{FL}}(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_{\mathsf{TV}}. \tag{1.30}$$

In an image denoising context, Rudin et al. (1992) proposed to minimize the total variation norm[21] *of the spatial derivatives of the corrupted image, and produced results that are much more visually pleasant. The fused LASSO regularizer is able to get the best of both ends, yielding estimates that not only contain many zeros but also whose nonzero elements are spatially smooth (in fact, piecewise constant). Note that the total variation norm is a "transformed" ℓ_1 in the sense that $\|\mathbf{w}\|_{\mathsf{TV}} = \left\|D^{[1,m]}\mathbf{w}\right\|_1$, where the matrix $D^{[1,m]} \in \mathbb{R}^{(m-1)\times m}$ is the first order difference operator (entries not shown are zeros)*[22]*:*

$$D^{[1,m]} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}.$$

Tibshirani et al. (2005) formulated the fused LASSO, under the least squares loss, as an instance of quadratic programming, and resorted to a generic solver. Later on Friedman et al. (2007) showed the surprising result:

$$\mathsf{P}^{\eta}_{f_{\mathsf{FL}}} = \mathsf{P}^{\eta}_{\lambda_1\|\cdot\|_1} \circ \mathsf{P}^{\eta}_{\lambda_2\|\cdot\|_{\mathsf{TV}}}, \tag{1.31}$$

[21] Rudin et al. (1992) considered *grouping* the spatial derivatives u_x, u_y at coordinate (x, y) as $\sqrt{u_x^2 + u_y^2}$ while another possibility is $|u_x| + |u_y|$ (Friedman et al. 2007).

[22] Hebiri and van de Geer (2011) analyzed the Smooth-LASSO regularizer $\lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \left\|D^{[1,m]}\mathbf{w}\right\|_2^2$, which, as pointed out in Tibshirani et al. (2005), might not lead to piecewise constant estimates.

where $\circ$ denotes the composition. Friedman et al. (2007) designed a coordinate-wise algorithm for $\mathsf{P}^{\eta}_{\|\cdot\|_{\mathsf{TV}}}$, which, unfortunately, does to extend to non-orthogonal designs. Zhang et al. (2013) mentioned an efficient dynamic programming algorithm for computing exactly $\mathsf{P}^{\eta}_{\|\cdot\|_{\mathsf{TV}}}$, see also Condat (2013).

Of course, we could consider sparsity in higher order differences by iterating:

$$\mathbb{R}^{(m-k)\times m} \ni D^{[k,m]} = D^{[1,m-k+1]} \circ D^{[k-1,m]}.$$

The regularizer $\left\|D^{[k,m]}\mathbf{w}\right\|_1$ encourages a piecewise $(k-1)$-degree polynomial estimate. In particular, Kim et al. (2009) considered $\left\|D^{[2,m]}\mathbf{w}\right\|_1$ in trend filtering, as an alternative to the traditional Hodrick-Prescott regularizer $\left\|D^{[2,m]}\mathbf{w}\right\|_2^2$. Exploiting the fact that the matrix $D^{[k,m]}$ is k-banded, Kim et al. (2009) developed a primal-dual interior point algorithm to compute the proximal map in $O(k^2m^{1.5})$.

Example 1.9 (Graph Sparsity, Hoefling (2010); Kim and Xing (2009))**.** *This is a generalization of Example 1.8. Instead of considering the differences between consecutive entries, we assume some "proximity" between the variables is available, such as an undirected graph (V, E) whose nodes V represent variables and whose edges E indicate "neighbors". The belief is that neighbors tend to have the same estimate, and naturally we penalize their differences in our estimation algorithm by*

$$\sum_{\{i,j\}\in E} \|\mathbf{w}_i - \mathbf{w}_j\|,$$

which is readily extended to the time series setting where $\mathbf{w}_i$ is the (vectorial) parameter for time slot i. Example 1.8 above is a special case with the graph being simply a chain. Of course, it is also easy to consider higher order differences. However, the proximal map is no longer easily computable. Kim and Xing (2009) thresholded the correlation matrix of the features to obtain the graph, and solved the regularized problem using a variational approach. Hoefling (2010) designed a homotopy algorithm for the proximal map with an unanalyzed complexity.

Example 1.10 (Matrix Sparsity, Candès and Recht (2009))**.** *This is a generalization of Example 1.4. We observe a small number of entries in some unknown matrix $Z \in \mathbb{R}^{m\times n}$, and our task is to infer the remaining entries. A popular application is the Netflix problem, where the rows of the matrix Z represent users and the columns represent movies. Each user can possibly rate only very few movies that he or she has watched, that is, we only get to observe a small portion of the matrix Z. The machine learner's job is to complete the matrix Z so as to provide (hopefully appreciated) recommendations for different users. Of course, other recommendation systems can be modeled more or less the same way. Needless to say that this is a highly ill-posed problem, and a reasonable and popular assumption is that the matrix Z is of very low rank. For instance, there are only a handful factors that affect users' preferences on movies. Unfortunately, the rank function is not convex, in fact NP-Hard to minimize even subject to linear constraints. Much like what we saw in Example 1.4,*

a convex relaxation that has been extremely useful in the current matrix setting is the trace norm $\|\cdot\|_{\mathrm{tr}}$*, defined as the sum of singular values—indeed the matrix version of the* ℓ_1 *norm. Putting things together, we arrive at the mathematical formulation of the matrix completion problem*

$$\min_{W\in\mathbb{R}^{m\times n}} \ell(\mathcal{P}(Z)-W)+\lambda\|W\|_{\mathrm{tr}}\,, \tag{1.32}$$

where $\mathcal{P}:\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$ *is the mask operator which simply fills the unobserved entries in* Z *with zero, and* ℓ *is some loss function which we choose to fit the observed entries. Surprisingly, Candès and Recht (2009) proved that under the low rank assumption, the solution of the convex relaxation* (1.32) *will uncover the true matrix* Z *with high probability, even though we only observed a very small* random *portion. Candès and Recht (2009) reformulated* (1.32) *as an instance of semidefinite programming (SDP) and resorted to a generic SDP solver which allowed them to handle matrices with sizes a few dozens by dozens—far from practically useful.*

Due to its apparent practical value, a lot of algorithms have thus been proposed to push the limit on the size of matrices to tens of thousands or even larger. PG (Ma et al. 2011) and APG (Pong et al. 2010; Toh and Yun 2010) are among those most promising algorithms, but they require access to the proximal map of the trace norm. Not too surprisingly, just like the ℓ_1 *norm for vectors, we do have again a "closed-form" solution. For any matrix* W*, let its singular value decomposition (SVD) be* $W=\sum_i \sigma_i\mathbf{u}_i\mathbf{v}_i^\top$ *with* $\{\sigma_i>0\}$ *being its singular values and* $\{\mathbf{u}_i\}$*,* $\{\mathbf{v}_i\}$ *being its left and right singular vectors, respectively, then*[23]

$$\mathsf{P}^{\eta}_{\|\cdot\|_{\mathrm{tr}}}(W):=\operatorname*{argmin}_X \tfrac{1}{2}\|X-W\|_F^2+\eta\|X\|_{\mathrm{tr}}=\sum_i(\sigma_i-\eta)_+\mathbf{u}_i\mathbf{v}_i^\top, \tag{1.33}$$

i.e. *we apply the soft-thresholding operator in* (1.21) *to the singular values and leave the singular vectors untouched. This result, a direct consequence of von Neumann (1937, Theorem 1), is formally proved by Cai et al. (2010); Ma et al. (2011) while a more general result is supplied in Yu and Schuurmans (2011). Unfortunately, performing a* full *SVD to get the proximal map in each iteration is exceedingly expensive, costing* $O(n^3)$ *for an* n *by* n *matrix. Existing proximal methods relied on heuristics to conduct a reduced SVD which can still be costly.*

1.7 Contributions

We have reviewed four popular gradient algorithms for optimizing the composite problem (1.1) whose presence in machine learning can be vividly felt. The efficiency of these algorithms (and some others) is completely determined by the proximal map of the regularizer, *cf.* (1.20). This proximal map is indeed available in closed-form for simple regularizers such as the ℓ_1 norm in LASSO (Example 1.4), the group norm in non-overlapping group LASSO (Example 1.6) and the total variation norm in fused LASSO (Example 1.8). However, for more complicated regularizers such as

[23] The Frobenius norm is defined as $\|W\|_F:=(\sum_i\sum_j W_{ij}^2)^{1/2}$.

those in the overlapping group LASSO (Example 1.7), graph-guided fused LASSO (Example 1.9), and matrix completion (Example 1.10), their proximal maps are not so easy to compute. Therefore an extra computational effort is needed, which is the focus of this thesis.

In Chapter 2, motivated by the decomposition results in Jenatton et al. (2011) and Friedman et al. (2007), we identify conditions under which the proximal map of a sum of functions is simply the composition of the proximal map of each individual summand. We not only give a unified treatment of existing known results, but also find several new decompositions. These results can be readily plugged into the four algorithms we reviewed in Chapter 1. An unexpected connection is the complete equivalence of one of our characterizations with the newly found characterization of the representer theorem in kernel methods.

Also shown in Chapter 2 is the negative result about the (frequent) failure of prox-decompositions, therefore quite naturally in Chapter 3 we look for approximations of the regularizer whose proximal map is troublesome. We restrict ourselves to regularizers which can be written as a sum of much simpler functions, as this seems to cover a lot of interesting regularizers in machine learning, *cf.* Example 1.7 and Example 1.9. Since regularizers are mostly nonsmooth functions, a generic way is to approximate them by some *smooth* function, for instance, the Moreau envelop that we saw in this chapter, and then apply gradient algorithms with no "explicit" nonsmooth component. Somewhat surprisingly, with all the advancement we have seen on nonsmooth optimization, such as the last three algorithms we reviewed, the dominating strategy is still to smooth any trouble-making regularizer, as if we could only handle smooth functions. We take a different, perhaps even naive at first glance, approach in Chapter 3, that is, we pretend that the proximal map is a linear operator, even though it apparently is not. This bold idea trivially makes algorithms like PG or APG applicable. Interestingly, through a new tool in convex analysis—the proximal average, we formally justify the resulting algorithms, with even a strictly better convergence rate than the usual smoothing strategy. Numerical experiments conducted on overlapping group LASSO (Example 1.7) and graph-guided fused LASSO (Example 1.9) corroborate our theoretical claims.

In Chapter 4, motivated by the matrix completion problem in Example 1.10, we present yet another algorithm called generalized conditional gradient (GCG), a generalization of the old conditional gradient due to Frank and Wolfe (1956). GCG is flexible enough to cover a lot of algorithms as special cases, for instance, RDA as we showed in Section 1.5. More importantly, unlike the four reviewed algorithms which require the proximal map of the regularizer, GCG requires what we call the polar operator, which is simply the dual norm if the regularizer is a norm. In some settings, the polar can be significantly cheaper than the proximal map. For instance, for the trace norm used in Example 1.10, its polar is the spectral norm whose computation only costs $O(n^2)$ for an n by n matrix, as opposed to the $O(n^3)$ complexity of its proximal map. We give a fairly complete overview of GCG and propose a variant that handles positively homogeneous regularizers, with special attention on establishing the convergence rate of GCG, which turns out to be on the same order as PG,

but slower than APG. To further accelerate convergence, we combine GCG with an efficient local search procedure, and demonstrate its superiority on some matrix applications. We also discuss the potential of GCG as a generic convex relaxation tool in dictionary learning.

We conclude with some discussions and future directions in Chapter 5.

Most results in this thesis have been published previously: Chapter 2 in Yu (2013b); Yu et al. (2013), Chapter 3 in Yu (2013a), and Chapter 4 in Zhang et al. (2012).

Chapter 2

Prox-decomposition

We saw in Chapter 1 that the proximal map is the key component in many gradient-type algorithms, which have become prevalent in large-scale high-dimensional applications. For simple functions this proximal map is available in closed-form while for more complicated functions it can become highly nontrivial. Motivated by the need of combining regularizers to simultaneously induce different types of structures, *e.g.* Example 1.7 and Example 1.9, in this chapter we systematically investigate when the proximal map of a sum of functions decomposes into the composition of the proximal maps of the individual summands. We not only unify a few known results scattered in the literature but also discover several new decompositions obtained almost effortlessly from our theory. An unexpected result is the connection with the representer theorem in kernel methods.

The results in this chapter appeared in Yu (2013b); Yu et al. (2013).

2.1 Introduction

We demonstrated the relevance of regularization in *e.g.* statistics, signal processing and machine learning, through a sequence of examples in Section 1.1. As real data become more and more complex, different types of regularizers, usually nonsmooth functions, have been designed. In many applications, it is thus desirable to combine regularizers, usually taking their sum, to promote different structures simultaneously.

Since many interesting regularizers are nonsmooth functions, they are harder to optimize numerically, especially in large-scale high-dimensional settings. This new challenge motivated the recent advances in nonsmooth optimization, in particular, gradient-type algorithms, whose per-iteration complexity is low, have been generalized to take regularizers explicitly into account; we discussed some of them in Chapter 1. The key component of many of these algorithms is the proximal map (of the nonsmooth regularizer), which is available in closed-form in simple settings. However, the proximal map becomes highly nontrivial when we start to combine regularizers.

The main goal of this chapter is to systematically investigate when the proximal map of a sum of functions decomposes into the composition of the proximal maps of the individual summands,

which we simply term prox-decomposition. Our motivation comes from a few known decomposition results scattered in the literature (Friedman et al. 2007; Jenatton et al. 2011; Zhou et al. 2012), all in the form of our interest. The study of such prox-decompositions is not only of mathematical interest, but also the backbone of popular gradient-type algorithms, such as those we reviewed in Chapter 1. More importantly, a precise understanding of this decomposition will shed light on how we should combine regularizers, taking computational efforts explicitly into account.

After setting the context in Section 2.2, we motivate the prox-decomposition with some justifications, as well as some cautionary results. Based on a sufficient condition presented in Section 2.3.1, we study how "invariance" of the subdifferential of one function would lead to nontrivial prox-decompositions. Specifically, we prove in Section 2.3.3 that when the subdifferential of one function is scaling invariant, then the prox-decomposition always holds if and only if another function is radial—which is, quite unexpectedly, exactly the same condition proven recently for the validity of the representer theorem in kernel methods (Dinuzzo and Schölkopf 2012; Yu et al. 2013). The generalization to cone invariance is considered in Section 2.3.4, and enables us to recover most known prox-decompositions, as well as some new ones falling out quite naturally. For completeness, Section 2.4 presents the related proof for the characterization of the representer theorem.

2.2 Proximal Map

Recall that our domain $\mathcal{H}$ is a (real) Hilbert space equipped with the inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\|\cdot\|_{\mathcal{H}}$. If needed, we will assume that some fixed orthonormal basis $\{\mathbf{e}_i\}_{i \in I}$ is chosen for $\mathcal{H}$, so that for $\mathbf{x} \in \mathcal{H}$ we are able to refer to its "coordinates" $x_i = \langle \mathbf{x}, \mathbf{e}_i \rangle$. As before Γ_0 denotes the set of all closed proper $\mathbb{R} \cup \{\infty\}$-valued convex functions on $\mathcal{H}$.

Fix the convex function $f \in \Gamma_0$. Moreau (1965) first studied the envelop function

$$\forall \mathbf{z} \in \mathcal{H}, \quad \mathsf{M}_f(\mathbf{z}) \;=\; \min_{\mathbf{x} \in \mathcal{H}} \tfrac{1}{2} \|\mathbf{x} - \mathbf{z}\|_{\mathcal{H}}^2 + f(\mathbf{x}), \tag{2.1}$$

and the related proximal map

$$\mathsf{P}_f(\mathbf{z}) \;=\; \operatorname*{argmin}_{\mathbf{x} \in \mathcal{H}} \tfrac{1}{2} \|\mathbf{x} - \mathbf{z}\|_{\mathcal{H}}^2 + f(\mathbf{x}). \tag{2.2}$$

The alert reader observes that we have slightly changed the definition, as compared to the one in (1.20). Indeed, the constant $\frac{1}{2}$ should be $\frac{1}{2\eta}$, to take the step size (or Lipschitz constant) into account. Here, to simplify the notation, we have chosen to absorb this constant to f, without loss of much generality. In fact, historically, our current definition is the one studied by Moreau (1965). As mentioned before, the proximal map is the key component of many gradient-type algorithms, such as those discussed in Chapter 1.

Since $f \in \Gamma_0$ and $\|\cdot\|_{\mathcal{H}}^2$ is strongly convex[1], the Moreau envelop and the proximal map are well-defined and single-valued[2]. This justifies the notations $\min$ and argmin in the above definition. Note

[1] Recall that f is σ-strongly convex iff $f - \frac{\sigma}{2} \|\cdot\|_{\mathcal{H}}^2$ is convex for some $\sigma > 0$.

[2] The argument is of the usual Weierstraß type: closed functions attain their minimum on compact sets.

that $\mathsf{M}_f : \mathcal{H} \to \mathbb{R}$ is real-valued while $\mathsf{P}_f : \mathcal{H} \to \mathcal{H}$ is not. Apply the optimality condition (1.11) to (2.2) we obtain

$$\mathsf{P}_f = (\mathsf{Id} + \partial f)^{-1}, \tag{2.3}$$

and

$$\mathsf{P}_f(\mathbf{z}) = \mathbf{z} \iff \mathbf{z} \in \operatorname{argmin} f. \tag{2.4}$$

Clearly, when $f = \iota_C$ is the indicator of some closed convex set C, the proximal map reduces to the usual Hilbertian projection. Interestingly, many (but not all) properties of the projection operator transfer to proximal maps. For instance, proximal maps are nonexpansive[3], just like projections. Perhaps the most interesting property of M_f, known as Moreau's identity, is the following decomposition (Moreau 1965)

$$\mathsf{M}_f(\mathbf{z}) + \mathsf{M}_{f^*}(\mathbf{z}) = \tfrac{1}{2} \|\mathbf{z}\|_{\mathcal{H}}^2 , \tag{2.5}$$

where f^* is the Fenchel conjugate of f, *cf.* Definition 1.3. Moreau (1965) proved that M_f is Frechét differentiable, hence taking derivative *w.r.t.* $\mathbf{z}$ in both sides of (2.5) yields

$$\mathsf{P}_f(\mathbf{z}) + \mathsf{P}_{f^*}(\mathbf{z}) = \mathbf{z}, \tag{2.6}$$

which is exactly the motivation for Moreau to generalize projections to proximal maps:

Proposition 2.1 (Moreau)**.** *Let $\mathcal{K}$ be a closed convex cone[4] and $\mathcal{K}^\circ := \{\mathbf{y} \in \mathcal{H} : \langle \mathbf{x}, \mathbf{y} \rangle \leq 0, \forall \mathbf{x} \in \mathcal{K}\}$ be its polar, then for all $\mathbf{z} \in \mathcal{H}$, the following are equivalent*

- $\mathbf{z} = \mathbf{x} + \mathbf{y}, \mathbf{x} \in \mathcal{K}, \mathbf{y} \in \mathcal{K}^\circ, \langle \mathbf{x}, \mathbf{y} \rangle = 0$;
- $\mathbf{x} = \mathsf{P}_{\mathcal{K}}(\mathbf{z}), \mathbf{y} = \mathsf{P}_{\mathcal{K}^\circ}(\mathbf{z})$.

Note that we have abused the notation $\mathsf{P}_{\mathcal{K}} = \mathsf{P}_{\iota_{\mathcal{K}}}$ a bit. Take $\mathcal{K}$ a closed subspace we recover the familiar orthogonal decomposition in linear algebra. We can also exploit the identity (2.6) to simplify the computation of the proximal map, since sometimes one of P_f and P_{f^*} is easier to handle than the other.

Example 2.1. *We mentioned in Example 1.4 the soft-thresholding operator $[\mathsf{P}_f(\mathbf{z})]_i = z_i(1 - 1/|z_i|)_+$, where $f = \|\cdot\|_1$. We now derive it through (2.6), although a direct calculation is not hard either. Indeed, by Cauchy-Schwarz we verify that $f^* = \iota_{\{\|\cdot\|_\infty \leq 1\}}$. Easily we compute $[\mathsf{P}_{f^*}(\mathbf{z})]_i = \operatorname{sign}(z_i) \cdot \min\{|z_i|, 1\}$. Appealing to (2.6) we obtain the claimed soft-thresholding operator.*

Quite remarkably, in the same paper, Moreau (1965) gave a complete characterization of proximal maps:

[3] Recall that a map $T : \mathcal{H} \to \mathcal{H}$ is nonexpansive if it is 1-Lipschitz continuous, that is, $\|T(\mathbf{x}) - T(\mathbf{y})\|_{\mathcal{H}} \leq \|\mathbf{x} - \mathbf{y}\|_{\mathcal{H}}$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{H}$.

[4] A set is a cone if it is invariant under positive scaling, *i.e.*, $\lambda \cdot \mathcal{K} = \mathcal{K}, \forall \lambda \geq 0$ (or $\lambda > 0$ for some authors).

Theorem 2.1 (Moreau (1965)). *$\mathsf{P} : \mathcal{H} \to \mathcal{H}$ is the proximal map of some function $f \in \Gamma_0$ if and only if it is nonexpansive and there exists $\mathsf{M} \in \Gamma_0$ such that $\forall \mathbf{z} \in \mathcal{H}, \mathsf{P}(\mathbf{z}) \in \partial \mathsf{M}(\mathbf{z})$.*

The downside though, is that the latter condition, that is, whether or not a given map is the subdifferential of a closed proper convex function, is hard to verify in general[5]. Some exceptions are summarized below.

Corollary 2.1. *The linear map $A : \mathcal{H} \to \mathcal{H}$ is a proximal map if and only if it is nonexpansive, self-adjoint and positive*[6].

Corollary 2.2. *The map $\mathsf{P} : \mathbb{R} \to \mathbb{R}$ is a proximal map if and only if it is nonexpansive and monotonically increasing.*

More properties of proximal maps will be presented in Chapter 3, and Proposition 2.4 below.

2.3 Decomposition

Our main goal is to investigate and understand the equality

$$\mathsf{P}_{f+g} \stackrel{?}{=} \mathsf{P}_f \circ \mathsf{P}_g \stackrel{?}{=} \mathsf{P}_g \circ \mathsf{P}_f, \tag{2.7}$$

where $f, g \in \Gamma_0$ and $f \circ g$ denotes the mapping composition. Out interest of (2.7) comes from combining say two regularizers f and g: (2.7) allows us to reduce the computation of P_{f+g} to a simple function of P_f and P_g, which themselves can be computed in many cases, as we will see. Note that Γ_0 is not convex, therefore $f+g$ might not be in Γ_0, making P_{f+g} undefined. We exclude this triviality, *i.e.* $f + g \equiv \infty$, in the whole chapter since it is clearly not of our interest.

Under the technical assumption[7] $\partial(f+g) = \partial f + \partial g$, and use (2.3),

$$\begin{aligned}\mathsf{P}_{f+g} &= (\mathsf{Id} + \partial(f+g))^{-1} = (\mathsf{Id} + \partial f + \partial g)^{-1} = \left[\frac{(\mathsf{Id} + 2\partial f) + (\mathsf{Id} + 2\partial g)}{2}\right]^{-1} \\ &= \left[\frac{\mathsf{P}_{2f}^{-1} + \mathsf{P}_{2g}^{-1}}{2}\right]^{-1} = (\mathsf{P}_{2f}^{-1} + \mathsf{P}_{2g}^{-1})^{-1} \circ (2\mathsf{Id}).\end{aligned} \tag{2.8}$$

However, computationally this formula is of little use. On the other hand, it is possible to develop forward-backward splitting procedures to numerically compute P_{f+g}, using only P_f and P_g as subroutines (Combettes et al. 2011). In some sense, this procedure is to compute $\mathsf{P}_{f+g} \approx \lim_{t\to\infty}(\mathsf{P}_f \circ \mathsf{P}_g)^t$, modulo some intermediate steps. Our focus is on the exact closed-form formula (2.7), essentially, establishing the one-step convergence of the iterative procedure of Combettes et al. (2011). Interestingly, under some "shrinkage" assumption, the prox-decomposition (2.7), even when it is *false*, can still be used in subgradient algorithms (Martins et al. 2011).

Our first result is encouraging:

[5] Another equivalent condition that we are aware of is the maximal cyclic monotonicity, which does not appear to be easy to verify either.

[6] Meaning that $\langle \mathbf{z}, A\mathbf{z} \rangle \geq 0$ for all $\mathbf{z} \in \mathcal{H}$.

[7] Note that the former always contains the latter while the reverse holds when, say one of the functions is continuous at some point in $\operatorname{dom} f \cap \operatorname{dom} g$, see Ekeland and Témam (1999, Proposition 5.6).

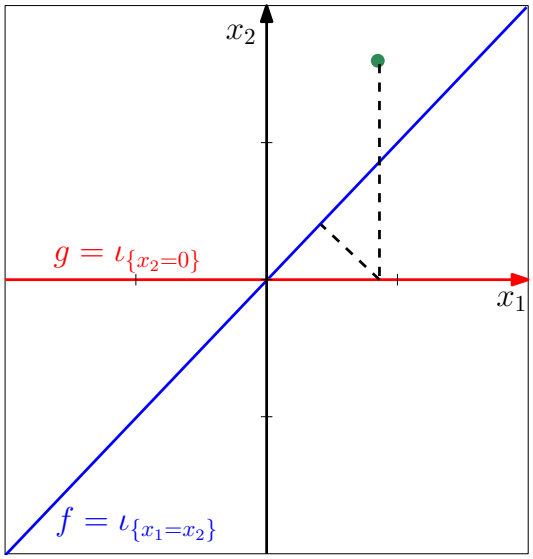


Figure 2.1: Composition of (linear) projections fails to be a proximal map.

Proposition 2.2. *If $\mathcal{H} = \mathbb{R}$, then for any $f, g \in \Gamma_0$, there exists $h \in \Gamma_0$ such that $\mathsf{P}_h = \mathsf{P}_f \circ \mathsf{P}_g$.*

Proof. Since both P_f and P_g are increasing and nonexpansive, it follows easily that so is $\mathsf{P}_f \circ \mathsf{P}_g$. By Corollary 2.2 there exists some $h \in \Gamma_0$ so that $\mathsf{P}_h = \mathsf{P}_f \circ \mathsf{P}_g$. □

In a general Hilbert space $\mathcal{H}$, we again easily conclude that the composition $\mathsf{P}_f \circ \mathsf{P}_g$ is always a nonexpansion, which means that it is "close" to be a proximal map. This justifies the composition $\mathsf{P}_f \circ \mathsf{P}_g$ as a candidate for the decomposition of P_{f+g}. However, we note that Proposition 2.2 indeed can fail already in $\mathbb{R}^2$:

Example 2.2. *Let $\mathcal{H} = \mathbb{R}^2$. Let $f = \iota_{\{x_1=x_2\}}$ and $g = \iota_{\{x_2=0\}}$. Clearly both f and g are in Γ_0. The proximal maps in this case are simply (linear) projections:*

$$\mathsf{P}_f = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \quad \mathsf{P}_g = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Therefore

$$\mathsf{P}_f \circ \mathsf{P}_g = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \end{bmatrix}$$

is a linear map that is not *self-adjoint, hence by Corollary 2.1 it is not a proximal map. It is also clear that any nontrivial scaling of $\mathsf{P}_f \circ \mathsf{P}_g$ cannot help either*[8]*. See Figure 2.1 for a pictorial illustration.*

Even worse, when Proposition 2.2 does hold, in general we can *not* expect the decomposition (2.7) to be true without additional assumptions.

Example 2.3. *Let $\mathcal{H} = \mathbb{R}$ and $\mathsf{q}(x) = \frac{1}{2}x^2$. It is easily seen that $\mathsf{P}_{\lambda\mathsf{q}}(x) = \frac{1}{1+\lambda}x$. Therefore $\mathsf{P}_\mathsf{q} \circ \mathsf{P}_\mathsf{q} = \frac{1}{4}\mathsf{Id} \neq \frac{1}{3}\mathsf{Id} = \mathsf{P}_{\mathsf{q}+\mathsf{q}}$. We will give an explanation for this failure of composition shortly.*

[8] In fact, proximal maps are firmly nonexpansive (Bauschke and Combettes 2011). Notice that for sufficiently small $\alpha > 0$, $\alpha \cdot \mathsf{P}_f \circ \mathsf{P}_g$ is firmly nonexpansive. Therefore even firm nonexpansions need not be proximal maps, even in $\mathbb{R}^2$.

Nevertheless, as we will see, the equality in (2.7) does hold in many scenarios, and an interesting theory can be suitably developed.

2.3.1 A Sufficient Condition

We start with a sufficient condition that yields (2.7). This result, although easy to obtain, will play a key role in our subsequent development.

Using the optimality condition (1.11) and the definition of the proximal map (2.2), we have

$$\mathsf{P}_{f+g}(\mathbf{z}) - \mathbf{z} + \partial(f+g)(\mathsf{P}_{f+g}(\mathbf{z})) \ni 0 \tag{2.9}$$

$$\mathsf{P}_{g}(\mathbf{z}) - \mathbf{z} + \partial g(\mathsf{P}_{g}(\mathbf{z})) \ni 0 \tag{2.10}$$

$$\mathsf{P}_{f}(\mathsf{P}_{g}(\mathbf{z})) - \mathsf{P}_{g}(\mathbf{z}) + \partial f(\mathsf{P}_{f}(\mathsf{P}_{g}(\mathbf{z}))) \ni 0. \tag{2.11}$$

Adding the last two equations we obtain

$$\mathsf{P}_{f}(\mathsf{P}_{g}(\mathbf{z})) - \mathbf{z} + \partial g(\mathsf{P}_{g}(\mathbf{z})) + \partial f(\mathsf{P}_{f}(\mathsf{P}_{g}(\mathbf{z}))) \ni 0. \tag{2.12}$$

Comparing (2.9) and (2.12) gives us a simple rule:

Theorem 2.2. *A sufficient condition for* $\mathsf{P}_{f+g}(\mathbf{z}) = \mathsf{P}_{f}\big(\mathsf{P}_{g}(\mathbf{z})\big)$ *for all* $\mathbf{z} \in \mathcal{H}$ *is that*

$$\forall\, \mathbf{y} \in \operatorname{dom} g,\ \partial g(\mathsf{P}_{f}(\mathbf{y})) \supseteq \partial g(\mathbf{y}). \tag{2.13}$$

Proof. Let $\mathbf{y} = \mathsf{P}_{g}(\mathbf{z})$. Then by (2.12) and the subdifferential rule $\partial(f+g) \supseteq \partial f + \partial g$ we verify that $\mathsf{P}_{f}(\mathsf{P}_{g}(\mathbf{z}))$ satisfies the optimality condition (2.9), hence follows $\mathsf{P}_{f+g}(\mathbf{z}) = \mathsf{P}_{f}\big(\mathsf{P}_{g}(\mathbf{z})\big)$ since the proximal map is single-valued. □

We note that a special form of our sufficient condition has appeared in the proof of Zhou et al. (2012, Theorem 1), whose main result also follows immediately from our Theorem 2.5 below. Let us fix f, and define

$$\mathcal{K}_f = \{g \in \Gamma_0 : f + g \not\equiv \infty, (f, g) \text{ satisfy (2.13)}\}. \tag{2.14}$$

This yields immediately the next result.

Proposition 2.3. *For any* $f \in \Gamma_0$, $\mathcal{K}_f$ *is a cone. Moreover, if* $g_1 \in \mathcal{K}_f, g_2 \in \mathcal{K}_f$, $f + g_1 + g_2 \not\equiv \infty$ *and* $\partial(g_1 + g_2) = \partial g_1 + \partial g_2$, *then* $g_1 + g_2 \in \mathcal{K}_f$ *too.*

The last condition $\partial(g_1 + g_2) = \partial g_1 + \partial g_2$ in Proposition 2.3 is purely technical; it is satisfied when, say one of g_1 and g_2 is continuous at a single, arbitrary point in $\operatorname{dom} g_1 \cap \operatorname{dom} g_2$ (Ekeland and Témam 1999, Proposition 5.6). For comparison purpose, we note that it is not clear how $\mathsf{P}_{f+g+h} = \mathsf{P}_{f} \circ \mathsf{P}_{g+h}$ would follow from $\mathsf{P}_{f+g} = \mathsf{P}_{f} \circ \mathsf{P}_{g}$ and $\mathsf{P}_{f+h} = \mathsf{P}_{f} \circ \mathsf{P}_{h}$. This is the main motivation to consider the sufficient condition (2.13), which also explains the next definition.

Definition 2.1. *We call* $f \in \Gamma_0$ *self-prox-decomposable (s.p.d.) if* $f \in \mathcal{K}_{\alpha f}$ *for all* $\alpha > 0$.

For any s.p.d. f, since $\mathcal{K}_f$ is a cone, $\beta f \in \mathcal{K}_{\alpha f}$ for all $\alpha, \beta \geq 0$. Consequently, $\mathsf{P}_{(\alpha+\beta)f} = \mathsf{P}_{\beta f} \circ \mathsf{P}_{\alpha f} = \mathsf{P}_{\alpha f} \circ \mathsf{P}_{\beta f}$.

Remark 2.1. *A weaker definition for s.p.d. is to require $f \in \mathcal{K}_f$, from which we conclude that $\beta f \in \mathcal{K}_f$ for all $\beta \geq 0$, in particular $\mathsf{P}_{(m+n)f} = \mathsf{P}_{nf} \circ \mathsf{P}_{mf} = \mathsf{P}_{mf} \circ \mathsf{P}_{nf}$ for all natural numbers m and n. The two definitions coincide for positively homogeneous functions. We have not been able to construct a function that satisfies this weaker definition but not the stronger one in Definition 2.1.*

Example 2.4. *We easily verify that all affine functions $\ell = \langle \cdot, \mathbf{a} \rangle + b$ are s.p.d., in fact, they are the only differentiable functions that are s.p.d., which explains why Example 2.3 must fail. Another trivial class of s.p.d. functions are projectors to closed convex sets. Also, univariate positively homogeneous convex functions are s.p.d., due to Theorem 2.5 below. Some multivariate s.p.d. functions are given in Remark 2.5 below.*

The next example shows that the sufficient condition (2.13) is not necessary.

Example 2.5. *Fix $\mathbf{w} \in \mathcal{H}, f = \iota_{\{\mathbf{w}\}}$, and $g \in \Gamma_0$ with full domain. Clearly for any $\mathbf{x} \in \mathcal{H}$, $\mathsf{P}_{f+g}(\mathbf{z}) = \mathbf{w} = \mathsf{P}_f[\mathsf{P}_g(\mathbf{z})]$. However, since $\mathbf{z}$ is arbitrary, $\partial g(\mathsf{P}_f(\mathbf{z})) = \partial g(\mathbf{w}) \not\supseteq \partial g(\mathbf{z})$ if g is not linear.*

If $\dim(\mathcal{H}) = 1$, we can let $f = \iota_{[a,b]}$ for some $b \geq a$ and, say $g(x) = \frac{1}{2}x^2$. Theorem 2.3 below proves that $\mathsf{P}_{f+g} = \mathsf{P}_f \circ \mathsf{P}_g$ always holds. Clearly, the sufficient condition (2.13) is necessary only for points in the interval $[a, b]$.

On the other hand, if f and g are differentiable, then we actually have equality in (2.13), which is clearly necessary in this case. Since convex functions are almost everywhere differentiable (in the interior of their domain), we expect the sufficient condition (2.13) to be necessary "almost everywhere" too.

Thus we see that the key for the decomposition (2.7) to hold is to let the proximal map of f and the subdifferential of g "interact well" in the sense of (2.13). Interestingly, both are fully equivalent to the function itself[9]:

Proposition 2.4 (Moreau (1965))**.** *Let $f, g \in \Gamma_0$. The following are equivalent:*

i). $f = g + c$ *for some* $c \in \mathbb{R}$*;*

ii). $\partial f \subseteq \partial g$*;*

iii). $\mathsf{P}_f = \mathsf{P}_g$ *.*

Proof. i) $\Rightarrow$ ii): This is clear.

ii) $\Rightarrow$ iii): From (2.3) $\mathsf{P}_f = (\mathsf{Id} + \partial f)^{-1}$, hence $\mathsf{P}_f \subseteq \mathsf{P}_g$. But both are single-valued and everywhere defined, therefore we have in fact equality.

[9] In essence, the equivalence of i) and ii) is the familiar result in calculus. It remains true in an arbitrary Banach space but could fail in, say, an incomplete inner product space.

iii) $\Rightarrow$ i): Note that P_f is in fact the derivative of M_{f^*}, therefore by integration $\mathsf{P}_f = \mathsf{P}_g$ implies that $\mathsf{M}_{f^*} = \mathsf{M}_{g^*} - c$ for some $c \in \mathbb{R}$. Conjugating we get $(\mathsf{M}_{f^*})^* = (\mathsf{M}_{g^*})^* + c$. But $(\mathsf{M}_{f^*})^* = f + \frac{1}{2}\|\cdot\|_{\mathcal{H}}^2$. Canceling the squared norm we obtain $f = g + c$. □

Due to the equivalence in Proposition 2.4, some properties of the proximal map will transfer to corresponding properties of the function f itself, and vice versa. The next result is easy to obtain, and appeared essentially in Combettes and Pesquet (2007).

Proposition 2.5. *Let $f \in \Gamma_0$ and $\mathbf{z} \in \mathcal{H}$ be arbitrary, then*

i). P_f is odd if and only if f is even;

ii). $\mathsf{P}_f(U\mathbf{z}) = U\mathsf{P}_f(\mathbf{z})$ for all orthonormal matrices U if and only if $f(U\mathbf{z}) = f(\mathbf{z})$ for all orthonormal matrices U;

iii). $\mathsf{P}_f(Q\mathbf{z}) = Q\mathsf{P}_f(\mathbf{z})$ for all permutation Q (under some fixed basis) if and only if f is permutation invariant, that is $f(Q\mathbf{z}) = f(\mathbf{z})$ for all permutation Q.

Proof. The if parts follow from direct calculation. For the only if part in, say i), we verify directly from (2.2) that $\mathsf{P}_{f(-\cdot)}(\mathbf{z}) = -\mathsf{P}_f(-\mathbf{z}) = \mathsf{P}_f(\mathbf{z})$. Applying Proposition 2.4 we know f is even.

The other two cases are proved similarly. □

In the following, we will put some invariance assumptions on the subdifferential of g and accordingly find the right family of f whose proximal map "respects" that invariance. This way we will meet (2.13) by construction, hence effortlessly enjoy the prox-decomposition (2.7).

2.3.2 No Invariance

To begin with, consider first the trivial case where no invariance on the subdifferential of g is assumed. This is equivalent as requiring (2.13) to hold for all $g \in \Gamma_0$. Not surprisingly, we end up with a trivial choice of f.

Theorem 2.3. *Fix $f \in \Gamma_0$. $\mathsf{P}_{f+g} = \mathsf{P}_f \circ \mathsf{P}_g$ for all $g \in \Gamma_0$ if and only if*

- $\dim(\mathcal{H}) \geq 2$*; $f \equiv c$, or $f = \iota_{\{\mathbf{w}\}} + c$ for some $c \in \mathbb{R}$ and $\mathbf{w} \in \mathcal{H}$;*
- $\dim(\mathcal{H}) = 1$ *and $f = \iota_C + c$ for some closed and convex set C and $c \in \mathbb{R}$.*

Proof. $\Leftarrow$: We remind that the implicit constraint $f + g \not\equiv \infty$ is always in force. We need only consider $\dim(\mathcal{H}) = 1$ as the other case is clear. By definition

$$\mathsf{P}_{f+g}(z) = \operatorname*{argmin}_{x \in C}\{h_z(x) := \tfrac{1}{2}(z-x)^2 + g(x)\}.$$

Setting the derivative of $h_z(x)$ to zero we obtain $x^* = \mathsf{P}_g(z)$. Crucially, we observe that the one dimensional convex function $h_z(x)$ is decreasing on $]\inf\{\operatorname{dom} g\}, x^*[$ and increasing on $]x^*, \infty[$.

Also $C = [a, b]$ is a closed interval. Therefore, if $a \le x^* \le b$, we have $\mathsf{P}_{f+g}(z) = x^*$; if $x^* \ge b$, $\mathsf{P}_{f+g}(z) = b$; and if $x^* \le a$, $\mathsf{P}_{f+g}(z) = a$. In all cases, we verify $\mathsf{P}_{f+g}(z) = \mathsf{P}_f(x^*) = \mathsf{P}_f(\mathsf{P}_g(z))$.

$\Rightarrow$: We first prove that f is constant on its domain even when g is restricted to indicators. Indeed, let $\mathbf{x} \in \operatorname{dom} f$ and take $g = \iota_{\{\mathbf{x}\}}$. Then $\mathbf{x} = \mathsf{P}_{f+g}(\mathbf{x}) = \mathsf{P}_f[\mathsf{P}_g(\mathbf{x})] = \mathsf{P}_f(\mathbf{x})$, meaning that $\mathbf{x} \in \operatorname{argmin} f$, *cf.* (2.4). Since $\mathbf{x} \in \operatorname{dom} f$ is arbitrary, f is constant on its domain. The case $\dim(\mathcal{H}) = 1$ is complete. We consider the other case where $\dim(\mathcal{H}) \ge 2$ and $\operatorname{dom} f$ contains at least two points. If $\operatorname{dom} f \ne \mathcal{H}$, there exists $\mathbf{z} \notin \operatorname{dom} f$ such that $\mathsf{P}_f(\mathbf{z}) = \mathbf{y}$ for some $\mathbf{y} \in \operatorname{dom} f$, and closed convex set $C \cap \operatorname{dom} f \ne \emptyset$ with $\mathbf{y} \notin C \ni \mathbf{z}$. Let $g = \iota_C$ we obtain $\mathsf{P}_{f+g}(\mathbf{z}) \in C \cap \operatorname{dom} f$ while $\mathsf{P}_f(\mathsf{P}_g(\mathbf{z})) = \mathsf{P}_f(\mathbf{z}) = \mathbf{y} \notin C$, contradiction. □

The fundamental difference between $\dim(\mathcal{H}) = 1$ and $\dim(\mathcal{H}) \ge 2$ is not accidental; we will see it again below. Moreover, we notice that the prox-decomposition (2.7) is not symmetric in f and g, also reflected in the next result:

Theorem 2.4. *Fix $g \in \Gamma_0$. $\mathsf{P}_{f+g} = \mathsf{P}_f \circ \mathsf{P}_g$ for all $f \in \Gamma_0$ if and only if g is a continuous affine function.*

Proof. $\Leftarrow$: If $g = \langle \cdot, \mathbf{a} \rangle + c$, then $\mathsf{P}_g(\mathbf{z}) = \mathbf{z} - \mathbf{a}$. Easy calculation reveals that $\mathsf{P}_{f+g}(\mathbf{z}) = \mathsf{P}_f(\mathbf{z} - \mathbf{a}) = \mathsf{P}_f[\mathsf{P}_g(\mathbf{z})]$.

$\Rightarrow$: The converse is true even when f is restricted to continuous linear functions. Indeed, let $\mathbf{a} \in \mathcal{H}$ be arbitrary and consider $f = \langle \cdot, \mathbf{a} \rangle$. Then $\mathsf{P}_{f+g}(\mathbf{z}) = \mathsf{P}_g(\mathbf{z} - \mathbf{a}) = \mathsf{P}_f(\mathsf{P}_g(\mathbf{z})) = \mathsf{P}_g(\mathbf{z}) - \mathbf{a}$. Letting $\mathbf{a} = \mathbf{z}$ yields $\mathsf{P}_g(\mathbf{z}) = \mathbf{z} + \mathsf{P}_g(\mathbf{0}) = \mathsf{P}_{\langle \cdot, -\mathsf{P}_g(\mathbf{0}) \rangle}(\mathbf{z})$. Since $\mathbf{z}$ is arbitrary, by Proposition 2.4 we know that g is equal to a continuous affine function. □

Naturally, the next step is to put invariance assumptions on the subdifferential of g, effectively restricting the function class of g. As a trade-off, the function class of f, that satisfies (2.13), becomes larger so that nontrivial results will arise.

2.3.3 Scaling Invariance

The first invariant property we consider is scaling-invariance. What kind of convex functions have their subdifferential invariant to (positive) scaling? Assuming $\mathbf{0} \in \operatorname{dom} g$ and by simple integration[10]

$$g(t\mathbf{z}) - g(\mathbf{0}) = \int_0^t g'(s\mathbf{z}) \mathrm{d}s = \int_0^t \langle \mathbf{z}, \partial g(s\mathbf{z}) \rangle \, \mathrm{d}s = t \cdot [g(\mathbf{z}) - g(\mathbf{0})],$$

where the last equality follows from the scaling invariance of the subdifferential of g. Therefore, up to some additive constant, g is positively homogeneous (p.h.). On the other hand, if $g \in \Gamma_0$ is p.h. (automatically $0 \in \operatorname{dom} g$), then from definition we verify that ∂g is scaling-invariant. Therefore, under the scaling-invariance assumption, g consists of all p.h. functions in Γ_0, up to some additive constant. Consequently, the requirement on f is to have its proximal map $\mathsf{P}_f(\mathbf{z}) = \lambda_{\mathbf{z}} \cdot \mathbf{z}$ for

[10] Here $g'(s\mathbf{z})$, as a function of the scalar s, denotes its right derivative, or, thanks to the convexity of g and the "robustness" of integration, any other sensible selection of the subdifferential.

some $\lambda_{\mathbf{z}} \in [0,1]$ that may depend on $\mathbf{z}$ as well[11]. The next theorem completely characterizes such functions.

Theorem 2.5. *Let $f \in \Gamma_0$. Consider the statements*

i). $f = h(\|\cdot\|_{\mathcal{H}})$ for some increasing function $h : \mathbb{R}_+ \to \mathbb{R} \cup \{\infty\}$;

ii). For all perpendicular $\mathbf{x} \perp \mathbf{y} \implies f(\mathbf{x}+\mathbf{y}) \geq f(\mathbf{y})$;

iii). For all $\mathbf{z} \in \mathcal{H}$, $\mathsf{P}_f(\mathbf{z}) = \lambda_{\mathbf{z}} \cdot \mathbf{z}$ for some $\lambda_{\mathbf{z}} \in [0,1]$;

iv). $\mathbf{0} \in \operatorname{dom} f$ and $\mathsf{P}_{f+\kappa} = \mathsf{P}_f \circ \mathsf{P}_\kappa$ for all p.h. (up to some additive constant) functions $\kappa \in \Gamma_0$.

Then we have i) $\implies$ ii) $\iff$ iii) $\iff$ iv). Moreover, when $\dim(\mathcal{H}) \geq 2$, ii) $\implies$ i) as well, in which case $\mathsf{P}_f(\mathbf{z}) = \mathsf{P}_h(\|\mathbf{z}\|_{\mathcal{H}})/\|\mathbf{z}\|_{\mathcal{H}} \cdot \mathbf{z}$ (where we interpret $0/0 = 0$).

Remark 2.2. *When $\dim(\mathcal{H}) = 1$, ii) is equivalent as requiring f to attain its minimum at 0, in which case the implication ii) $\implies$ iv), under the redundant condition that f is differentiable, was proved by Combettes and Pesquet (2007, Proposition 3.6). The implication ii) $\implies$ iii) also generalizes Combettes and Pesquet (2007, Corollary 2.5), where only the case $\dim(\mathcal{H}) = 1$ and f differentiable was considered. Note that there exists non-even f that satisfies Theorem 2.5 when $\dim(\mathcal{H}) = 1$. Such is impossible for $\dim(\mathcal{H}) \geq 2$, in which case any f that satisfies Theorem 2.5 must also enjoy all properties listed in Proposition 2.5.*

Proof. i) $\implies$ ii): For perpendicular vectors $\mathbf{x} \perp \mathbf{y}$, we have $\|\mathbf{x}+\mathbf{y}\|_{\mathcal{H}} \geq \|\mathbf{y}\|_{\mathcal{H}}$.

ii) $\implies$ iii): Fix $\mathbf{z} \in \mathcal{H}$. For $\mathbf{x} \in \mathcal{H}$, let $\mathbf{x} = \lambda\mathbf{z} + \mathbf{z}^\perp$ be its orthogonal decomposition. By definition

$$\begin{aligned}
\mathsf{M}_f(\mathbf{z}) &= \min_{\mathbf{x}} \tfrac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 + f(\mathbf{x}) \\
&= \min_{\mathbf{z}^\perp,\lambda} \tfrac{1}{2}\|\mathbf{z}^\perp + \lambda\mathbf{z} - \mathbf{z}\|^2 + f(\mathbf{z}^\perp + \lambda\mathbf{z}) \\
&= \min_{\lambda} \tfrac{1}{2}\|\lambda\mathbf{z} - \mathbf{z}\|^2 + f(\lambda\mathbf{z}) \\
&= \min_{\lambda\in[0,1]} \tfrac{1}{2}(\lambda-1)^2\|\mathbf{z}\|^2 + f(\lambda\mathbf{z}),
\end{aligned}$$

where the third equality is due to ii), and the additional constraints on λ in the last equality can be seen as follows: For any $\lambda < 0$, by increasing it to 0 we can only decrease both terms; similar argument for $\lambda > 1$. Therefore there exists $\lambda_{\mathbf{z}} \in [0,1]$ such that $\lambda_{\mathbf{z}}\mathbf{z}$ minimizes the Moreau envelop M_f hence we have $\mathsf{P}_f(\mathbf{z}) = \lambda_{\mathbf{z}}\mathbf{z}$ due to uniqueness.

iii) $\implies$ iv): Note first that from iii) we have $\mathsf{P}_f(\mathbf{0}) = \mathbf{0}$, implying $\mathbf{0} \in \partial f(\mathbf{0})$ hence $\mathbf{0} \in \operatorname{dom} f$. Since the subdifferential of κ is scaling-invariant, iii) implies the sufficient condition (2.13) hence iv).

[11] Note that $\lambda_{\mathbf{z}} \leq 1$ is necessary since any proximal map is nonexpansive.

iv) $\Longrightarrow$ iii): Fix $\mathbf{z} \in \operatorname{dom} f$ and construct the p.h. convex function

$$\kappa(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} = \lambda \cdot \mathbf{z} \text{ for some } \lambda \geq 0 \\ \infty, & \text{otherwise} \end{cases}.$$

Then $\mathsf{P}_\kappa(\mathbf{z}) = \mathbf{z}$, hence $\mathsf{P}_f(\mathsf{P}_\kappa(\mathbf{z})) = \mathsf{P}_f(\mathbf{z}) = \mathsf{P}_{f+\kappa}(\mathbf{z})$ by iv). On the other hand,

$$\begin{aligned} \mathsf{M}_{f+\kappa}(\mathbf{z}) &= \min_{\mathbf{x}} \tfrac{1}{2} \|\mathbf{x} - \mathbf{z}\|_{\mathcal{H}}^2 + f(\mathbf{x}) + \kappa(\mathbf{x}) \\ &= \min_{\lambda \geq 0} \tfrac{1}{2} \|\lambda \mathbf{z} - \mathbf{z}\|_{\mathcal{H}}^2 + f(\lambda \mathbf{z}). \end{aligned} \tag{2.15}$$

Take $\mathbf{z} = \mathbf{0}$ we obtain $\mathsf{P}_{f+\kappa}(\mathbf{0}) = \mathbf{0}$. Thus $\mathsf{P}_f(\mathbf{0}) = \mathbf{0}$, *i.e.* $\mathbf{0} \in \partial f(\mathbf{0})$, from which we deduce that $\mathsf{P}_f(\mathbf{z}) = \mathsf{P}_{f+\kappa}(\mathbf{z}) = \lambda \mathbf{z}$ for some $\lambda \in [0, 1]$, since $f(\lambda \mathbf{z})$ in (2.15), as a convex function of λ, is increasing on $[1, \infty[$.

iii) $\Longrightarrow$ ii): First note that iii) implies that $\mathsf{P}_f(\mathbf{0}) = \mathbf{0}$ hence $\mathbf{0} \in \partial f(\mathbf{0})$, in particular, $\mathbf{0} \in \operatorname{dom} f$. If $\dim(\mathcal{H}) = 1$ we are done, so we assume $\dim(\mathcal{H}) \geq 2$ in the rest of the proof. In this case, by Theorem 2.7 below we know that ii) is equivalent as i), even without assuming f convex. All we left is to prove iii) $\Longrightarrow$ ii) or equivalently i), for the case $\dim(\mathcal{H}) \geq 2$.

We first prove the case when $\operatorname{dom} f = \mathcal{H}$. By iii), $\mathsf{P}_f(\mathbf{z}) = \lambda_{\mathbf{z}} \mathbf{z}$ for some $\lambda_{\mathbf{z}} \in [0, 1]$. Using the optimality condition (1.11) for the proximal map we have $0 \in \lambda_{\mathbf{z}} \mathbf{z} - \mathbf{z} + \partial f(\lambda_{\mathbf{z}} \mathbf{z})$, that is $(\frac{1}{\lambda_{\mathbf{z}}} - 1)\mathbf{y} \in \partial f(\mathbf{y})$ for each $\mathbf{y} = \lambda_{\mathbf{z}} \mathbf{z} \in \operatorname{Range}(\mathsf{P}_f) = \mathcal{H}$, due to our assumption $\operatorname{dom} f = \mathcal{H}$. Now for any perpendicular vectors $\mathbf{x} \perp \mathbf{y}$, by the definition of the subdifferential,

$$f(\mathbf{x} + \mathbf{y}) \geq f(\mathbf{y}) + \langle \mathbf{x}, \partial f(\mathbf{y}) \rangle = f(\mathbf{y}) + \left\langle \mathbf{x}, (\tfrac{1}{\lambda_{\mathbf{z}}} - 1)\mathbf{y} \right\rangle = f(\mathbf{y}).$$

Note that when $\lambda_{\mathbf{z}} = 0$, $\mathbf{y} = \mathbf{0}$ and the above inequality still holds.

For the case when $\operatorname{dom} f \subset \mathcal{H}$, we consider the proximal average (Bauschke et al. 2008; Moreau 1965)

$$g = \mathsf{A}(f, \mathsf{q}) = [(\tfrac{1}{2}(f^* + \mathsf{q})^* + \tfrac{1}{4}\mathsf{q})^* - \mathsf{q}]^*, \tag{2.16}$$

where $\mathsf{q} = \frac{1}{2} \|\cdot\|_{\mathcal{H}}^2$. The somewhat peculiar formula in the above definition can be derived later when we discuss the proximal average more thoroughly in Chapter 3. Here, we exploit two nice properties of the proximal average: Firstly, since q is defined on the whole space, the proximal average g has full domain too (Bauschke et al. 2008, Corollary 4.7); Secondly, $\mathsf{P}_g(\mathbf{z}) = \frac{1}{2}\mathsf{P}_f(\mathbf{z}) + \frac{1}{4}\mathbf{z} = (\frac{1}{2}\lambda_{\mathbf{z}} + \frac{1}{4})\mathbf{z}$. Therefore by our previous argument, g satisfies ii) hence also i). It is easy to check that i) is preserved under taking the Fenchel conjugation (note that the convexity of f implies that of h). Since we have shown that g satisfies i), it follows by repeatedly conjugating (2.16) that f satisfies i) hence also ii).

As mentioned, when $\dim(\mathcal{H}) \geq 2$, the implication ii) $\Longrightarrow$ i) will be proven in Theorem 2.7 below. The formula $\mathsf{P}_f(\mathbf{z}) = \mathsf{P}_h(\|\mathbf{z}\|_{\mathcal{H}})/\|\mathbf{u}\|_{\mathcal{H}} \cdot \mathbf{z}$ for $f = h(\|\cdot\|_{\mathcal{H}})$ follows from straightforward calculation. □

Remark 2.3. *The idea behind the proof for iii) $\Longrightarrow$ ii) in Theorem 2.5 seems worth reiterating: The main difficulty is the subdifferentiability of the function f at points on the boundary of its effective*

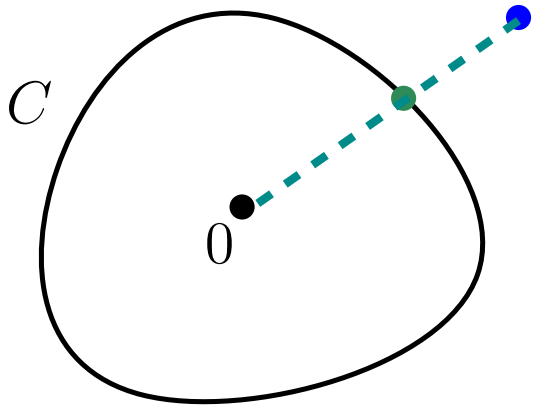


Figure 2.2: Characterization of the "roundness" of the Hilbertian ball.

domain. However, if the property we are interested in, such as being a function of the norm $\|\cdot\|_{\mathcal{H}}$, is preserved under taking Fenchel conjugation and is also enjoyed by, say the quadratic function $\mathsf{q} = \frac{1}{2}\|\cdot\|_{\mathcal{H}}^2$, we can assume without loss of generality that f has full domain, for otherwise we just consider its proximal average with q*—a* bona fide *convex function that is defined everywhere. This frees us from considering "unfriendly" points, and repeated conjugating can bring us back to f, without harming the property we are interested in. We expect this simple trick to have more applications.*

We now discuss some applications of Theorem 2.5. When $\dim(\mathcal{H}) \geq 2$, iii) in Theorem 2.5 automatically implies that the scalar constant $\lambda_{\mathbf{z}}$ depends on $\mathbf{z}$ only through its norm. This fact, although not entirely obvious, does have a clear geometric picture, as shown in Figure 2.2 and formalized below.

Corollary 2.3. *Let $\dim(\mathcal{H}) \geq 2$, $C \subseteq \mathcal{H}$ be a closed convex set that contains the origin. Then the projection of any point onto C is always a shrinkage towards the origin* (i.e.*, lying somewhere on the line segment connecting the point and the origin) if and only if C is a ball (of the norm $\|\cdot\|_{\mathcal{H}}$).*

Proof. The slight complication is that different points, even with the same length, may shrink to the origin with varying degrees. Excluding this possibility is not entirely trivial.

Let $f = \iota_C$ and apply Theorem 2.5. □

Example 2.6. *As usual, denote $\mathsf{q} = \frac{1}{2}\|\cdot\|_{\mathcal{H}}^2$. In many applications, in addition to the regularizer κ (usually a p.h. convex function), one adds the squared ℓ_2 regularizer $\lambda\mathsf{q}$ for stability, grouping effect, strong convexity,* etc. *This incurs no computational cost in the sense of computing the proximal map: We easily compute that $\mathsf{P}_{\lambda\mathsf{q}} = \frac{1}{\lambda+1}\mathsf{Id}$. By Theorem 2.5, for any p.h. convex function κ, $\mathsf{P}_{\kappa+\lambda\mathsf{q}} = \frac{1}{\lambda+1}\mathsf{P}_{\kappa}$, whence it is also clear that adding an extra squared ℓ_2 regularizer tends to double "shrink" the solution. In particular, let $\mathcal{H} = \mathbb{R}^d$ and take κ to be the ℓ_1 norm, we recover the proximal map for the elastic-net regularizer proposed by Zou and Hastie (2005).*

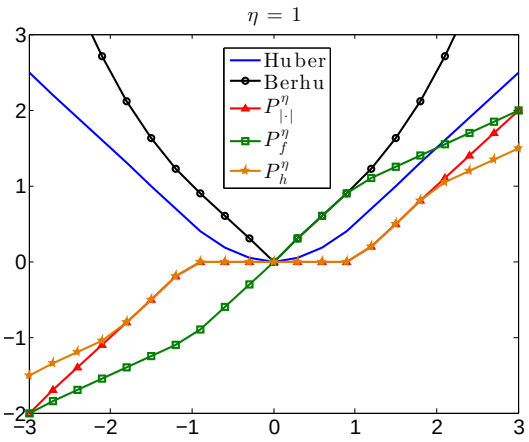


Figure 2.3: The proximal map of the Berhu regularizer.

Example 2.7. *Let $\mathcal{H} = \mathbb{R}$. The Berhu regularizer (with parameter $\gamma > 0$)*

$$h(x) = |x|\mathbf{1}_{|x|<\gamma} + \tfrac{x^2+\gamma^2}{2\gamma}\mathbf{1}_{|x|\geq\gamma} = |x| + \tfrac{(|x|-\gamma)^2}{2\gamma}\mathbf{1}_{|x|\geq\gamma}, \tag{2.17}$$

*being the reverse (even in its name!) of Huber's function (*cf. *Example 1.4), is proposed in Owen (2007) as a bridge between the lasso (ℓ_1 regularization) and ridge regression (squared ℓ_2 regularization). Let $f(x) = h(x) - |x|$. Clearly, f satisfies ii) of Theorem 2.5 (but not differentiable), hence*

$$\mathsf{P}_h = \mathsf{P}_f \circ \mathsf{P}_{|\cdot|},$$

whereas simple calculation verifies that

$$\mathsf{P}_f(x) = \operatorname{sign}(x) \cdot \min\{|x|, \tfrac{\gamma}{1+\gamma}(|x|+1)\},$$

and of course $\mathsf{P}_{|\cdot|}(x) = \operatorname{sign}(x) \cdot \max\{|x|-1, 0\}$. See Figure 2.3 for an illustration. Note that this regularizer is not s.p.d.

Corollary 2.4. *Let $\dim(\mathcal{H}) \geq 2$, then the p.h. function $f \in \Gamma_0$ satisfies any item of Theorem 2.5 if and only if it is a positive multiple of the norm $\|\cdot\|_{\mathcal{H}}$.*

Proof. Theorem 2.8 below showed that under positive homogeneity, i) in Theorem 2.5 implies that f is a positive multiple of the norm. □

Therefore, (positive multiples of) the Hilbertian norm is the only p.h. convex function f that satisfies $\mathsf{P}_{f+\kappa} = \mathsf{P}_f \circ \mathsf{P}_\kappa$ for all p.h. convex functions κ. In particular, this means that the norm $\|\cdot\|_{\mathcal{H}}$ is s.p.d. (*cf.* Definition 2.1). Moreover, we easily recover the following result that is perhaps not so obvious at first glance:

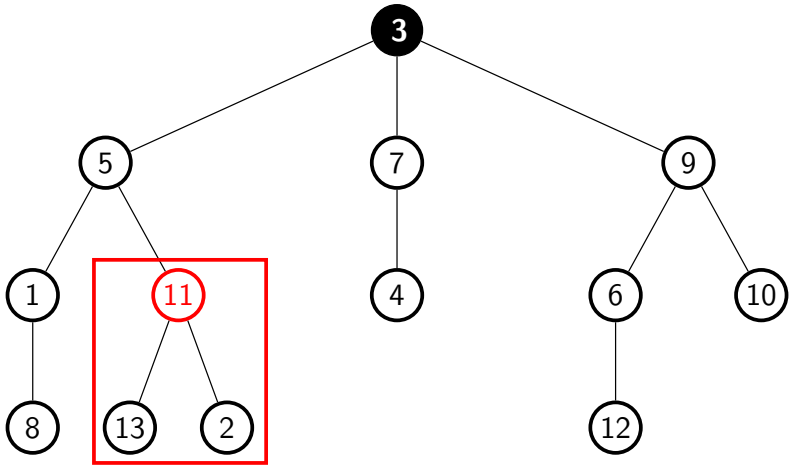


Figure 2.4: Tree-structured groups are simply rooted subtrees in a rooted tree. The red rectangle denotes the group induced by the subtree rooted at the red node.

Corollary 2.5 (Jenatton et al. (2011))**.** *Fix the orthonormal basis* $\{\mathbf{e}_i\}_{i\in I}$ *of* $\mathcal{H}$*. Let* $\mathcal{G} \subseteq 2^I$ *be a collection of tree-structured groups, that is, either* $g \subseteq g'$ *or* $g' \subseteq g$ *or* $g \cap g' = \emptyset$ *for all* $g, g' \in \mathcal{G}$*. Then*

$$\mathsf{P}_{\sum_{i=1}^n \|\cdot\|_{g_i}} = \mathsf{P}_{\|\cdot\|_{g_1}} \circ \cdots \circ \mathsf{P}_{\|\cdot\|_{g_n}},$$

where we arrange the groups so that $g_i \subset g_j \implies i > j$*, and the notation* $\|\cdot\|_{g_i}$ *denotes the Hilbertian norm that is restricted to the subspace spanned by the variables in group* g_i*.*

Proof. Let $f = \|\cdot\|_{g_1}$ and $\kappa = \sum_{i=2}^n \|\cdot\|_{g_i}$. Clearly they are both p.h. (and convex). By the tree-structured assumption we can partition $\kappa = \kappa_1 + \kappa_2$, where $g_i \subset g_1$ for all g_i appearing in κ_1 while $g_j \cap g_1 = \emptyset$ for all g_j appearing in κ_2. Restricting to the subspace spanned by the variables in g_1 we can treat f as the Hilbertian norm. Apply Theorem 2.5 we obtain $\mathsf{P}_{f+\kappa_1} = \mathsf{P}_f \circ \mathsf{P}_{\kappa_1}$. On the other hand, due to the non-overlapping property, it follows from an easy calculation that $\mathsf{P}_{(f+\kappa_1)+\kappa_2} = \mathsf{P}_{f+\kappa_1} \circ \mathsf{P}_{\kappa_2}$, thus a similar reasoning yields

$$\mathsf{P}_{\sum_{i=1}^n \|\cdot\|_{g_i}} = \mathsf{P}_{\|\cdot\|_{g_1}} \circ \mathsf{P}_{\sum_{i=2}^n \|\cdot\|_{g_i}}.$$

We can clearly iterate the argument to unravel the proximal map as claimed. □

For notational clarity, we have chosen not to incorporate weights in the sum of group seminorms: Such can be absorbed into the seminorm and the corollary clearly remains intact. Our proof also reveals the fundamental reason why Corollary 2.5 is true: The Hilbertian norm admits the prox-decomposition (2.7) for any p.h. convex function g! This fact, to the best of our knowledge, has not been recognized previously.

Note that the tree-structured set system $\mathcal{G}$ in Corollary 2.5 is called *laminar* in combinatorics. The name "tree-structured" comes from the fact that we can always rearrange the variables to sit

in a rooted tree (or forest more generally) so that the groups in $\mathcal{G}$ are simply rooted subtrees, see Figure 2.4 and Korte and Vygen (2012, Proposition 2.14). Somewhat disappointingly, the number of groups in a laminar system is at most twice the number of variables (Korte and Vygen 2012, Corollary 2.15), therefore tree-structured groups are not substantially more powerful than non-overlapping groups (whose size can be the number of variables).

2.3.4 Cone Invariance

In the previous section, we restricted the subdifferential of g to be constant along each ray. We now generalize this to cones. Specifically, consider the gauge, that is, a p.h. convex function

$$\kappa(\mathbf{x}) = \max_{j \in J} \langle \mathbf{a}_j, \mathbf{x} \rangle \,, \tag{2.18}$$

where J is a finite index set and each $\mathbf{a}_j \in \mathcal{H}$. Such (polyhedral) gauge functions have become extremely important in machine learning due to the work of Chandrasekaran et al. (2012). Define the polyhedral cones[12]

$$K_j = \{\mathbf{x} \in \mathcal{H} : \langle \mathbf{a}_j, \mathbf{x} \rangle = \kappa(\mathbf{x})\}. \tag{2.19}$$

Assume $K_j \neq \emptyset$ for each j (otherwise delete j from J). The sufficient condition (2.13), with $g = \kappa$, becomes $\partial\kappa(\mathsf{P}_f(\mathbf{y})) \supseteq \partial\kappa(\mathbf{y})$. Since $\partial\kappa(\mathbf{x}) = \{\mathbf{a}_j : j \in J, \mathbf{x} \in K_j\}$, $\mathbf{a}_j \in \partial\kappa(\mathbf{y}) \iff \mathbf{y} \in K_j$, hence $\mathbf{a}_j \in \partial\kappa(\mathsf{P}_f(\mathbf{y})) \iff \mathsf{P}_f(\mathbf{y}) \in K_j$. In other words, we simplify the sufficient condition (2.13) as

$$\forall j \in J, \;\; \mathsf{P}_f(K_j) \subseteq K_j \iff K_j \subseteq K_j + \partial f(K_j). \tag{2.20}$$

That is, each cone K_j is "fixed" under the proximal map of f. Instead of completely characterizing f under (2.20), we show that in its current form, (2.20) already implies many known results, with some new generalizations falling out naturally.

Corollary 2.6. *Denote E a collection of pairs $\{m, n\}$, and recall from Example 1.8 the total variational (semi)norm $\|\mathbf{x}\|_{\mathsf{TV}} = \sum_{\{m,n\} \in E} w_{mn} \cdot |x_m - x_n|$, where $w_{mn} \geq 0$. Then for any permutation invariant function[13] f, we have*

$$\mathsf{P}_{f + \|\cdot\|_{\mathsf{TV}}} = \mathsf{P}_f \circ \mathsf{P}_{\|\cdot\|_{\mathsf{TV}}}.$$

Proof. Pick an arbitrary pair $\{m, n\} \in E$ and let $\kappa = |x_m - x_n|$. Clearly $J = \{1, 2\}$, $K_1 = \{x_m \geq x_n\}$ and $K_2 = \{x_m \leq x_n\}$. Since f is permutation invariant, its proximal map $\mathsf{P}_f(\mathbf{x})$ maintains the relative order of entries in $\mathbf{x}$, see Proposition 2.5, hence we establish (2.20). The other way to get (2.20) is to verify, simply from the definition, that the subdifferential of a permutation invariant function is itself permutation invariant. Finally apply Proposition 2.3 and Theorem 2.2. □

[12] A set is polyhedral if it is the intersection of *finitely* many half spaces. Polyhedral sets are closed convex.

[13] Recall from Proposition 2.5 that f is permutation invariant if for all permutation matrix P we have $f(P\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x}$. Note that all we need is the weaker condition: For all $\{m, n\} \in E$, $x_m \geq x_n \implies [\mathsf{P}_f(\mathbf{x})]_m \geq [\mathsf{P}_f(\mathbf{x})]_n$.

The special case where f is the ℓ_1 norm, appeared first in Friedman et al. (2007), see Example 1.8. The generalization to any ℓ_p norm appeared in Zhang et al. (2013).

We call the permutation invariant function f symmetric if for all $\mathbf{x}$, $f(|\mathbf{x}|) = f(\mathbf{x})$, where $|\cdot|$ denotes the componentwise absolute value. The proof for the next corollary is almost the same as that of Corollary 2.6, except that we also use the fact $\operatorname{sign}([\mathsf{P}_f(\mathbf{x})]_i) = \operatorname{sign}(x_i)$ for symmetric functions (or the fact that the subdifferential of a symmetric function is itself symmetric).

Corollary 2.7. *As in Corollary 2.6, define the (semi)norm*

$$\|\mathbf{x}\|_{\mathsf{oct}} = \sum_{\{m,n\}\in E} w_{mn} \cdot \max\{|x_m|, |x_n|\}.$$

Then for any symmetric function f, $\mathsf{P}_{f+\|\cdot\|_{\mathsf{oct}}} = \mathsf{P}_f \circ \mathsf{P}_{\|\cdot\|_{\mathsf{oct}}}$.

Remark 2.4. *This norm $\|\cdot\|_{\mathsf{oct}}$ is proposed in Bondell and Reich (2008) for feature grouping, for it tends to pull x_m and x_n together for each $\{m,n\} \in E$. Surprisingly, Corollary 2.7 appears to be new. When the underlying graph of E is complete (and for simplicity let $w \equiv 1$), the proximal map $\mathsf{P}_{\|\cdot\|_{\mathsf{oct}}}$ is derived in Zhong and Kwok (2011), which turns out to be another decomposition result. Indeed, for $i \geq 2$, define $\kappa_i(\mathbf{x}) = \sum_{j\leq i-1} \max\{|x_i|, |x_j|\}$. Thus*

$$\|\cdot\|_{\mathsf{oct}} = \sum_{i\geq 2} \kappa_i.$$

Importantly, we observe that κ_i is symmetric on the first $i-1$ coordinates. We claim that

$$\mathsf{P}_{\|\cdot\|_{\mathsf{oct}}} = \mathsf{P}_{\kappa_{|I|}} \circ \ldots \circ \mathsf{P}_{\kappa_2}.$$

The proof is by recursion: Write $\|\cdot\|_{\mathsf{oct}} = f + g$, where $f = \kappa_{|I|}$ (recall that $|I|$ is the dimensionality of $\mathbf{x} \in \mathcal{H}$). Note that the subdifferential of g depends only on the ordering and sign of the first $|I|-1$ coordinates while the proximal map of f preserves the ordering and sign of the first $|I|-1$ coordinates (due to symmetry). If we pre-sort $\mathbf{x}$, the individual proximal maps $\mathsf{P}_{\kappa_i}(\mathbf{x})$ become easy to compute sequentially and we recover the algorithm in Zhong and Kwok (2011) after some bookkeeping.

Corollary 2.8. *As in Corollary 2.5, let $\mathcal{G} \subseteq 2^I$ be a collection of tree-structured groups, then*

$$\mathsf{P}_{\sum_{i=1}^n \|\cdot\|_{g_i,k}} = \mathsf{P}_{\|\cdot\|_{g_1,k}} \circ \cdots \circ \mathsf{P}_{\|\cdot\|_{g_n,k}},$$

where we arrange the groups so that $g_i \subset g_j \implies i > j$, and $\|\mathbf{x}\|_{g_i,k} = \sum_{j=1}^k |x_{g_i}|_{[j]}$ is the sum of the k (absolute-value) largest elements in the group g_i, i.e., *Ky-Fan's k-norm.*

Proof. Similar as in the proof of Corollary 2.5, we need only prove that

$$\mathsf{P}_{\|\cdot\|_{g_1,k}+\|\cdot\|_{g_2,k}} = \mathsf{P}_{\|\cdot\|_{g_1,k}} \circ \mathsf{P}_{\|\cdot\|_{g_2,k}},$$

where w.l.o.g. we assume g_1 contains all variables while $g_2 \subset g_1$. Therefore $\|\cdot\|_{g_1,k}$ can be treated as symmetric. To be explicit, let group $g_2 = \{i_1, \ldots, i_s\}$. Ky-Fan's k-norm on g_2 induces $m = \binom{s}{k}$,

or $m = 1$ if $s < k$, polyhedral cones $K_j, j = 1, \ldots, m$. Since $f = \|\cdot\|_{g_1,k}$ is symmetric, $\mathsf{P}_f(\mathbf{y})$ maintains the relative order of magnitudes in $\mathbf{y}$. In other words, $\mathsf{P}_f(K_j) \subseteq K_j$, establishing (2.20). Applying Theorem 2.2 completes the proof. □

Note that the case $k \in \{1, |I|\}$ was proved in Jenatton et al. (2011) and Corollary 2.8 can be seen as an interpolation. Interestingly, there is another interpolated result whose proof should be apparent now.

Corollary 2.9. *Corollary 2.8 remains true if we replace Ky-Fan's k-norm with*

$$\|\mathbf{x}\|_{\mathsf{oct},k} = \sum_{1 \le i_1 < i_2 < \ldots < i_k \le |I|} \max\{|x_{i_1}|, \ldots, |x_{i_k}|\}. \tag{2.21}$$

Therefore we can employ the norm $\|\mathbf{x}\|_{\mathsf{oct},2}$ for feature grouping in a hierarchical manner. Clearly we can also combine Corollary 2.8 and Corollary 2.9. Our last result does not bring any new technique but leads to important algorithmic consequences.

Corollary 2.10. *For any symmetric f, $\mathsf{P}_{f+\|\cdot\|_{\mathsf{oct},k}} = \mathsf{P}_f \circ \mathsf{P}_{\|\cdot\|_{\mathsf{oct},k}}$. Similarly, for Ky-Fan's k-norm $\|\mathbf{x}\|_k = \sum_{i=1}^k |x|_{[i]}$, we have $\mathsf{P}_{f+\|\cdot\|_k} = \mathsf{P}_f \circ \mathsf{P}_{\|\cdot\|_k}$.*

Remark 2.5. *Immediately, Corollary 2.10 implies that Ky-Fan's k-norm and the norm $\|\cdot\|_{\mathsf{oct},k}$ defined in (2.21) are both s.p.d. (see Definition 2.1). The special case for the ℓ_p norm with $p \in \{1, 2, \infty\}$ was proved in Duchi and Singer (2009, Proposition 11), with a substantially more complicated argument. As pointed out in Duchi and Singer (2009), s.p.d. regularizers allow us to perform lazy updates in PG (*cf. *Section 1.3) or PSG (*cf. *Section 1.4). Indeed, suppose during the iterate that the (sub)gradient of the loss ℓ is sparse, we need to perform the update* w.r.t. *the regularizer f by $\mathbf{w} \leftarrow \mathsf{P}_f^{\eta_t}(\mathbf{w})$. For those coordinates with (constantly) null (sub)gradient, instead of performing the proximal map in each step, we could just aggregate them in one-shot: $\mathbf{w} \leftarrow \mathsf{P}_f^{\sum_t \eta_t}(\mathbf{w})$, provided that f is s.p.d. Notice that the ℓ_p norm for other p is not s.p.d., as can be quickly verified by numerical examples, or see Jenatton et al. (2011) for a proof.*

Of course, we have not exhausted the possibility to have the prox-decomposition (2.7). For instance, all of our results extend to matrix variables, provided that we consider only unitarily invariant matrix norms, see Appendix A or Yu and Schuurmans (2011) for some relevant discussions. In our development (and the existing results we are aware of), we heavily build upon the "round" ℓ_2 norm or *polyhedral* functions[14]. Whether or not this is a sheer coincidence requires some further work on understanding the prox-decomposition (2.7).

2.4 Connection with the Representer Theorem

The main goal in this section is to supply the missing piece in the proof of Theorem 2.5, and draw the connection to the representer theorem in kernel methods. Some background first.

[14] A (convex) function f is polyhedral iff its epigraph $\{(\mathbf{x}, t) \in \mathcal{H} \times \mathbb{R} : f(\mathbf{x}) \le t\}$ is a polyhedral set.

Many kernel methods can be formulated as the optimization problem

$$\inf_{\mathbf{w}\in\mathcal{H}} \ell_n(\langle \mathbf{w}, \mathbf{w}_1\rangle, \ldots, \langle \mathbf{w}, \mathbf{w}_n\rangle) + f(\mathbf{w}), \tag{2.22}$$

where $\ell_n : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is some loss function, $\mathbf{w}_i \in \mathcal{H}, i = 1, \ldots, n$ is our data, and $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is the regularizer. Unfortunately, $\mathcal{H}$ is usually an infinite dimensional Hilbert space, thus optimizing (2.22) directly might run into practical issues. However, if we are assured, by a proper design of the regularizer f, that some minimizer actually lies in the span of the data, that is $\mathbf{w} = \sum_{i=1}^n \alpha_i \mathbf{w}_i$ for $\alpha_i \in \mathbb{R}, i = 1, \ldots, n$, we can turn (2.22) into a finite dimensional problem which simply finds $\boldsymbol{\alpha} \in \mathbb{R}^n$. Of course, we need to be able to compute the Gram matrix $K_{ij} = \langle \mathbf{w}_i, \mathbf{w}_j\rangle$. Such is the case when $\mathcal{H}$ is the reproducing kernel Hilbert space induced by some kernel function that is explicitly evaluable, see Aronszajn (1950) for details. Any regularizer f that enables the outlined reduction is said to satisfy the representer theorem. As a simple consequence of orthogonal decompositions in Hilbert space, any increasing function of the norm $\|\cdot\|_{\mathcal{H}}$, in particular $\|\cdot\|_{\mathcal{H}}^2$, satisfies the representer theorem (Kimeldorf and Wahba 1971; Scholköpf and Smola 2001). The quest is to supply a necessary condition hence completely characterize such regularizers.

As pointed out in Argyriou et al. (2009), to study the representer theorem, one can (and perhaps should) focus on the interpolation problem:

$$\inf_{\mathbf{w}\in\mathcal{H}} f(\mathbf{w}) \quad \text{s.t. } \langle \mathbf{w}, \mathbf{w}_i\rangle = y_i,\ i = 1, \ldots, n. \tag{2.23}$$

The advantage of considering interpolation is that the loss function ℓ_n no longer plays any role in the specification. Moreover, it is easy to argue that if f satisfies the representer theorem in (2.23), it remains so in (2.22) for any loss ℓ_n. The converse is also true, under minor regularity conditions on the loss ℓ_n, see Argyriou et al. (2009); Dinuzzo and Schölkopf (2012).

To facilitate the discussion, following Argyriou et al. (2009), we define the term *admissibility* as follows:

Definition 2.2. *The function $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is admissible if for all n, $(\mathbf{w}_i \in \mathcal{H})_{i=1}^n$ and $(y_i \in \mathbb{R})_{i=1}^n$, some minimizer of (2.23) admits the form*

$$\mathbf{w} = \sum_{i=1}^n \alpha_i \mathbf{w}_i \tag{2.24}$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^n$. We consider the statement vacuously true if (2.23) has no minimizer.

The key step towards characterizing admissible functions is due to Argyriou et al. (2009):

Proposition 2.6. *Let $\mathcal{H}$ be an inner product space. The function $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is admissible if and only if*

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{H}, \langle \mathbf{x}, \mathbf{y}\rangle = 0 \ \Rightarrow\ f(\mathbf{x} + \mathbf{y}) \geq f(\mathbf{x}). \tag{2.25}$$

Proof. $\Rightarrow$: Suppose f is admissible. Consider the following instance of (2.23):

$$\inf_{\mathbf{w}\in\mathcal{H}} f(\mathbf{w}) \text{ s.t. } \langle \mathbf{w}, \mathbf{x}\rangle = \|\mathbf{x}\|_{\mathcal{H}}^2 . \tag{2.26}$$

The admissibility of f implies that $\mathbf{x}$ is a minimizer of (2.26). Since $\mathbf{x}+\mathbf{y}$, for any $\mathbf{y} \perp \mathbf{x}$, is feasible for (2.26), we have $f(\mathbf{x}+\mathbf{y}) \geq f(\mathbf{x})$ from the optimality of $\mathbf{x}$.

$\Leftarrow$: Suppose (2.25) holds and (2.23) has a minimizer $\mathbf{z} = \mathbf{w} + \mathbf{w}^{\perp}$, where $\mathbf{w} \in \mathcal{H}_n := \operatorname{span}\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ and $\mathbf{w}^{\perp}$ is in the orthogonal complement of $\mathcal{H}_n$.[15] Clearly $\langle \mathbf{z}, \mathbf{w}_i\rangle = \langle \mathbf{w}, \mathbf{w}_i\rangle = y_i$, hence $\mathbf{w}$ is feasible. Invoking (2.25) we know $f(\mathbf{z}) \geq f(\mathbf{w})$, therefore $\mathbf{w}$ is also a minimizer, proving the admissibility of f. □

Although Proposition 2.6 gives a complete characterization of admissibility, the verification of its conditions can be cumbersome. Argyriou et al. (2009) further proved that for differentiable f, it is admissible if and only if it is an increasing of the norm $\|\cdot\|_{\mathcal{H}}$. Dinuzzo and Schölkopf (2012) managed to weaken the differentiability assumption to lower semicontinuity (l.s.c.)[16]. We now demonstrate that a modification of their proof removes even the l.s.c. requirement hence yields a complete characterization of admissibility.

We first make an easy observation. If the vector space $\mathcal{H}$ has unit dimension, *i.e.* $\dim(\mathcal{H}) = 1$, then the condition (2.25) is equivalent as requiring $f(\mathbf{w}) \geq f(\mathbf{0})$ for all $\mathbf{w} \in \mathcal{H}$. Therefore, for the remainder of this section we will exclude this trivial case and assume $\dim(\mathcal{H}) \geq 2$ henceforth.

The main result in this section, which, in retrospect could be considered to be the "correct form" of the representer theorem, is the following:

Theorem 2.6. *Let $\mathcal{H}$ be an inner product space with $\dim(\mathcal{H}) \geq 2$. A function $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is admissible if and only if*

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{H}, \|\mathbf{y}\|_{\mathcal{H}} > \|\mathbf{x}\|_{\mathcal{H}} \Rightarrow f(\mathbf{y}) \geq f(\mathbf{x}). \tag{2.27}$$

Note that we do *not* require any assumption, such as l.s.c., on f, and (2.27) is not the usual "increasing" property, but instead a weaker requirement—we henceforth refer to it as the *weakly increasing property*. Then, the condition equivalent to admissibility can be stated concisely as weakly increasing *w.r.t.* the norm $\|\cdot\|_{\mathcal{H}}$.

Proof. $\Leftarrow$: Suppose (2.27) holds. We verify (2.25), from which the admissibility of f will follow. Pick any $\mathbf{x}, \mathbf{y} \in \mathcal{H}$ such that $\langle \mathbf{x}, \mathbf{y}\rangle = 0, \mathbf{x} \neq 0$. Then we have $\|\mathbf{x}+\mathbf{y}\|_{\mathcal{H}} > \|\mathbf{y}\|_{\mathcal{H}}$ and thus by (2.27), $f(\mathbf{x}+\mathbf{y}) \geq f(\mathbf{y})$. Noting that the case $\mathbf{x} = 0$ also trivially holds, we see that (2.25) holds. By Proposition 2.6, we get that f is admissible.

[15] The existence of such a decomposition depends only on the completeness of $\mathcal{H}_n$, not on that of $\mathcal{H}$. Note that $\mathcal{H}_n$ is indeed complete since it is of finite dimension.

[16] Recall that $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is lower semicontinuous iff its sublevel set $\{\mathbf{w} \in \mathcal{H} : f(\mathbf{w}) \leq \alpha\}$ is closed for all $\alpha \in \mathbb{R}$.

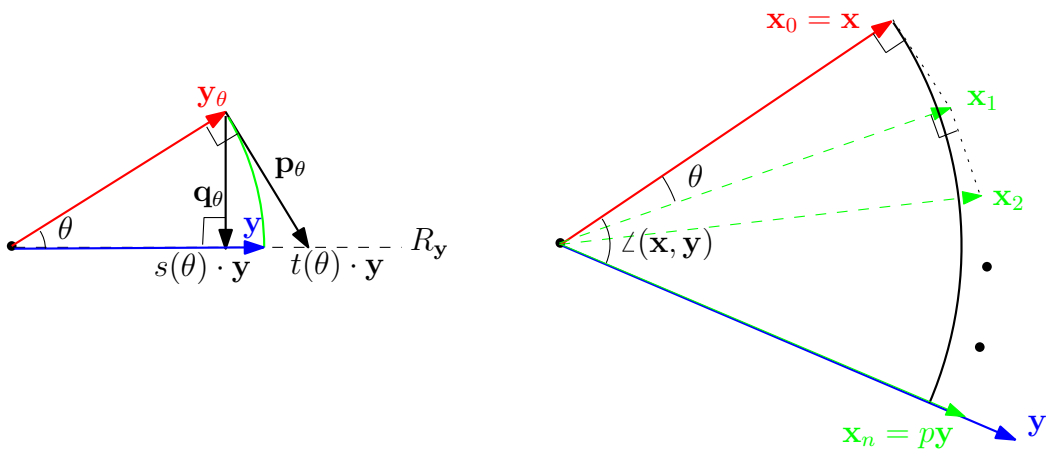


Figure 2.5: Illustration of the main idea presented in the proof of Theorem 2.6.

$\Rightarrow$: Suppose now that f is admissible. Then, by Proposition 2.6, (2.25) holds. Note that in the special case when $\mathbf{x}=0$ and $\mathbf{y}\neq 0$ (so that $\|\mathbf{y}\|_\mathcal{H}>0$), we have $f(\mathbf{y})=f(\mathbf{0}+\mathbf{y})\geq f(\mathbf{0})=f(\mathbf{x})$. Therefore, in what follows we need only deal with the case when $\mathbf{x}\neq 0$. To prove (2.27), we start with a claim.

Claim: The admissibility of f implies $f(\cdot)$ is increasing along any ray $R_\mathbf{y}=\{t\cdot\mathbf{y}\,:\,t\geq 0\}$, where $\mathbf{0}\neq\mathbf{y}\in\mathcal{H}$.

By the above reasoning it suffices to prove this claim for $R_\mathbf{y}\setminus\{\mathbf{0}\}$. We prove the claim using a geometric argument depicted in the left panel of Figure 2.5. For a fixed vector $\mathbf{y}\in\mathcal{H}$ and an angle $\theta\in[0,\pi/2[$, choose some $\mathbf{x}\in\mathcal{H}$ such that $\mathbf{x}$ is not parallel to $\mathbf{y}$. Such an $\mathbf{x}$ exists since $\dim(\mathcal{H})\geq 2$. Now, let $\mathbf{y}_\theta$ be the rotation of $\mathbf{y}$ in the plane (subspace) P spanned by $\mathbf{x}$ and $\mathbf{y}$. The direction of rotation can be chosen arbitrarily. Take the line in the plane P that passes through $\mathbf{y}_\theta$ and which is orthogonal to $\mathbf{y}_\theta$. Let $t(\theta)\cdot\mathbf{y}$ be the point where the ray $R_\mathbf{y}$ and the line intersect and let the vector $\mathbf{p}_\theta$ be defined as $\mathbf{y}_\theta+\mathbf{p}_\theta=t(\theta)\cdot\mathbf{y}$. Note that $t(\theta)=(1+\tan^2(\theta))^{1/2}\geq 1$ for all $\theta\in[0,\pi/2[$. Thus, $\mathbf{p}_\theta$ is orthogonal to $\mathbf{y}_\theta$: $\mathbf{p}_\theta\perp\mathbf{y}_\theta$. Further, let $s(\theta)\cdot\mathbf{y}$ be the orthogonal projection of $\mathbf{y}_\theta$ to the ray $R_\mathbf{y}$ and call $\mathbf{q}_\theta$ the vector that satisfies $s(\theta)\cdot\mathbf{y}+\mathbf{q}_\theta=\mathbf{y}_\theta$. Thus, $\mathbf{q}_\theta\perp s(\theta)\cdot\mathbf{y}$. Further, $s(\theta)=\cos(\theta)\leq 1$ for all $\theta\in[0,\pi/2[$. Applying (2.25) from Proposition 2.6 twice we get

$$\begin{aligned} f(t(\theta)\cdot\mathbf{y}) &= f(\mathbf{y}_\theta+\mathbf{p}_\theta)\geq f(\mathbf{y}_\theta) \\ &= f(s(\theta)\cdot\mathbf{y}+\mathbf{q}_\theta)\geq f(s(\theta)\cdot\mathbf{y}). \end{aligned} \tag{2.28}$$

Note that this holds for any $\mathbf{0}\neq\mathbf{y}\in\mathcal{H}$ and $\theta\in[0,\pi/2[$.

Now, take any $0<\tau_1<\tau_2$. Since $t(\theta)/s(\theta)$ is continuous on $[0,\pi/2[$ and its range is $[1,\infty[$, there exists a value $\theta'\in[0,\pi/2[$ such that

$$\frac{t(\theta')}{s(\theta')}=\frac{\tau_2}{\tau_1}\,. \tag{2.29}$$

Define $c=\tau_2/t(\theta')$. So we also have that $c=\tau_1/s(\theta')$ thanks to (2.29). Hence, applying (2.28) to

$c\mathbf{y}$ and θ', we get

$$f(\tau_2\mathbf{y}) = f\big(t(\theta')\cdot(c\mathbf{y})\big) \geq f\big(s(\theta')\cdot(c\mathbf{y})\big) = f(\tau_1\mathbf{y})\,,$$

finishing the proof of the claim.

Now if $\|\mathbf{y}\|_{\mathcal{H}} > \|\mathbf{x}\|_{\mathcal{H}}$ and $\mathbf{x}$ is not aligned with $\mathbf{y}$, it is not hard to see (cf. Figure 2.5, right panel) that one can find a sufficiently large $n \geq 1$, a real number $p \in]0,1[$ and a sequence $\mathbf{x}_0 = \mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_n = p\mathbf{y}$ such that for any $0 \leq i \leq n-1$, the angle $\angle(\mathbf{x}_i, \mathbf{x}_{i+1}) = \theta := \angle(\mathbf{x},\mathbf{y})/n$ and $(\mathbf{x}_{i+1} - \mathbf{x}_i) \perp \mathbf{x}_i$. Indeed, n defines the above sequence uniquely with some $p = p_n > 0$. In particular, $p_n \|\mathbf{y}\|_{\mathcal{H}} = \|\mathbf{x}_n\|_{\mathcal{H}} = [t(\theta/n)]^n \|\mathbf{x}\|_{\mathcal{H}}$, so $p_n = [t(\theta/n)]^n \frac{\|\mathbf{x}\|_{\mathcal{H}}}{\|\mathbf{y}\|_{\mathcal{H}}}$. Since $[t(\theta/n)]^n \sim (1+(\theta/n)^2)^{n^2/\theta^2\cdot\theta^2/n} \sim e^{\theta^2/n} \to 1$ as $n\to\infty$, $p_n \to \frac{\|\mathbf{x}\|_{\mathcal{H}}}{\|\mathbf{y}\|_{\mathcal{H}}} < 1$ and so the existence of (n,p) with the said properties is guaranteed. Therefore, using the claim and (2.25), we get

$$\begin{aligned} f(\mathbf{y}) &\geq f(p\mathbf{y}) \\ &= f(\mathbf{x}_n) = f(\mathbf{x}_{n-1} + (\mathbf{x}_n - \mathbf{x}_{n-1})) \\ &\geq f(\mathbf{x}_{n-1}) = f(\mathbf{x}_{n-2} + (\mathbf{x}_{n-1} - \mathbf{x}_{n-2})) \\ &\quad\vdots \\ &\geq f(\mathbf{x}_0) = f(\mathbf{x})\,, \end{aligned}$$

thus finishing the proof of (2.27). □

The reason why the continuity conditions can be avoided in Theorem 2.6, making the result simpler and more elegant, is that the necessary condition for the admissibility of f avoids stipulating f's behavior on the surface of balls. In fact, if one modified (2.27) to include the case when $\|\mathbf{x}\|_{\mathcal{H}} = \|\mathbf{y}\|_{\mathcal{H}}$, it would imply that f is *radial*, *i.e.*, $f(\mathbf{x})$ depends on the argument $\mathbf{x}$ only through $\|\mathbf{x}\|_{\mathcal{H}}$. The next example demonstrates that one can have an admissible regularizer that is not radial (of course, such an f cannot be semicontinuous).

Example 2.8. *Figure 2.6 shows an admissible function f that is not radial. The gray area denotes, say, the unit ball $\{\mathbf{w}\in\mathcal{H} : \|\mathbf{w}\|_{\mathcal{H}} \leq 1\}$ and the red point represents some $\mathbf{y}$ on the unit sphere $\{\mathbf{w}\in\mathcal{H} : \|\mathbf{w}\|_{\mathcal{H}} = 1\}$. It is clear that f is neither l.s.c. nor upper semicontinuous*[17]*. Note also that f is in fact a convex admissible function, demonstrating that convex functions can be "ugly" on boundary points.*

Remark 2.6. *As the previous example demonstrates, there exist non-radial, but admissible functions. However, Theorem 2.6 also implies that every admissible function is equal to an admissible radial function except for a set whose cardinality is at most "countable". To see this consider the function $I(r) := \inf\{f(\mathbf{x}) : \|\mathbf{x}\|_{\mathcal{H}} = r\}$. Clearly $I : \mathbb{R}_+ \to \mathbb{R}\cup\{\infty\}$ is an increasing function, hence it can have at most countably many discontinuity points. But it is easily seen that for any*

[17] Similarly, f is upper semicontinuous (u.s.c.) if its superlevel set $\{\mathbf{w}\in\mathcal{H} : f(\mathbf{w}) \geq \alpha\}$ is closed for all $\alpha\in\mathbb{R}$.

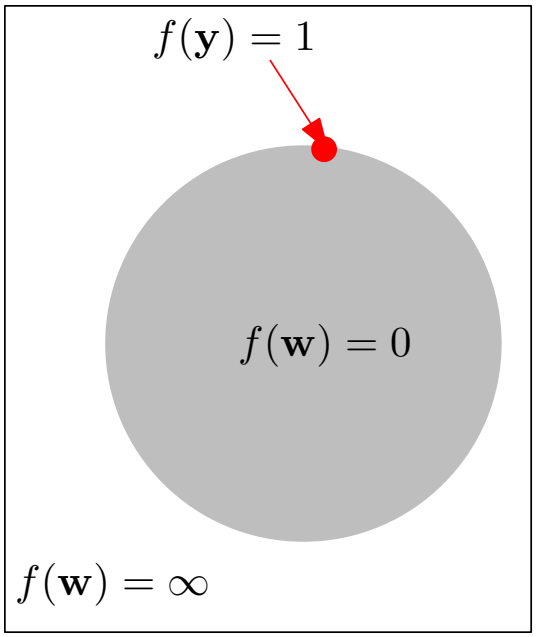


Figure 2.6: An admissible function f that is *not* increasing *w.r.t.* the norm.

continuity point r of I and any $\mathbf{x}, \mathbf{y} \in \mathcal{H}$ on the $\mathcal{H}$-sphere of radius r, it follows that $f(\mathbf{x}) = f(\mathbf{y})$. Thus, f is radial except for at most countably many spheres.

Before refining Theorem 2.6, let us mention that a function $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is u.s.c. iff for all $\mathbf{x} \in \mathcal{H}$ and the sequence $\mathbf{x}_n \to \mathbf{x}$, $f(\mathbf{x}) \geq \limsup_{\mathbf{x}_n \to \mathbf{x}} f(\mathbf{x}_n)$; similar result holds for l.s.c. functions, with $\limsup$ replaced by $\liminf$ and $\geq$ replaced by $\leq$. Of course, f is continuous iff it is both l.s.c. and u.s.c.

Remark 2.7. *One should not confuse the l.s.c. (u.s.c.) of $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ with the l.s.c. (u.s.c.) of $f : \operatorname{dom} f \to \mathbb{R}$. The former condition, used throughout this thesis, is strictly stronger than the latter condition. For instance, the f in Figure 2.6 is u.s.c. in the latter sense but not u.s.c. in our standard.*

We are now ready to provide the missing piece in the proof of Theorem 2.5. Obviously, any item in Theorem 2.5 also gives a different characterization of the representer theorem (under the l.s.c. assumption).

Theorem 2.7. *Let $\mathcal{H}$ be an inner product space with $\dim(\mathcal{H}) \geq 2$ and $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be u.s.c or l.s.c., then f is admissible if and only if*

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{H}, \|\mathbf{y}\|_{\mathcal{H}} \geq \|\mathbf{x}\|_{\mathcal{H}} \;\Rightarrow\; f(\mathbf{y}) \geq f(\mathbf{x}), \tag{2.30}$$

or, in other words, f is an increasing radial function.

Proof. $\Leftarrow$: (2.30) apparently implies (2.27) hence the admissibility of f.

$\Rightarrow$: Assume that f is u.s.c. and admissible. Thanks to Theorem 2.6, we need only prove that if $\|\mathbf{y}\|_{\mathcal{H}} = \|\mathbf{x}\|_{\mathcal{H}}$ then $f(\mathbf{y}) \geq f(\mathbf{x})$. To see this, take a sequence $\mathbf{y}_n$ that converges to $\mathbf{y}$ and that satisfies $\|\mathbf{y}_n\|_{\mathcal{H}} > \|\mathbf{y}\|_{\mathcal{H}}$. Then, $\|\mathbf{y}_n\|_{\mathcal{H}} > \|\mathbf{y}\|_{\mathcal{H}} = \|\mathbf{x}\|_{\mathcal{H}}$ also holds; therefore, by Theorem 2.6, $f(\mathbf{y}_n) \geq f(\mathbf{x})$ holds for all n. Taking the $\limsup$ of both sides, we get $f(\mathbf{y}) \geq \limsup_{n\to\infty} f(\mathbf{y}_n) \geq f(\mathbf{x})$.

The l.s.c. case can be proved using an entirely analogous argument, which is essentially the main result of Dinuzzo and Schölkopf (2012). Note that we cannot naively negate an l.s.c. function here to reduce to the u.s.c. case, since our starting tool (2.25) is *not* invariant to negation. □

Another easy way to see the result in Theorem 2.7 is to notice that the function $I(r)$ defined in Remark 2.6 is in fact continuous when f satisfies (2.30) (or equivalently (2.25)) and is either l.s.c. or u.s.c.

It turns out that positive homogeneity, other than semicontinuity, also forces admissible functions to be radial. Notice that both properties imply that the function $I(r)$ defined in Remark 2.6 is continuous.

Theorem 2.8. *Let $\mathcal{H}$ be an inner product space with $\dim(\mathcal{H}) \geq 2$. If f is admissible and positively homogeneous, then it is a positive multiple of the induced norm $\|\cdot\|_{\mathcal{H}}$.*

Proof. We prove first that f must be an increasing function of the norm. Note that due to positive homogeneity, we have $f(\mathbf{0}) = 0$ hence $f \geq 0$ by the admissibility. Suppose to the contrary there exist $\mathbf{x}, \mathbf{y} \in \mathcal{H}$ such that $\|\mathbf{x}\|_{\mathcal{H}} = \|\mathbf{y}\|_{\mathcal{H}} \neq 0$ but $f(\mathbf{x}) > f(\mathbf{y})$. Clearly $f(\mathbf{y}) < \infty$. Then for all $1 < \lambda < f(\mathbf{x})/f(\mathbf{y})$, $\|\lambda\mathbf{y}\|_{\mathcal{H}} = \lambda \|\mathbf{y}\|_{\mathcal{H}} > \|\mathbf{x}\|_{\mathcal{H}}$, hence $f(\lambda\mathbf{y}) \geq f(\mathbf{x})$ by the admissibility. If $f(\mathbf{y}) = 0$ then due to positive homogeneity $0 \geq f(\mathbf{x})$, contradiction; similarly, if $f(\mathbf{y}) > 0$, due to again positive homogeneity, $\lambda \geq f(\mathbf{x})/f(\mathbf{y})$, contradiction again. Thus f is an increasing radial function.

Take an arbitrary $\mathbf{x}_0 \in \operatorname{dom} f$ with unit norm (*i.e.*, $\|\mathbf{x}_0\|_{\mathcal{H}} = 1$), then due to positive homogeneity $f(\mathbf{x}) = \|\mathbf{x}\|_{\mathcal{H}} \cdot f(\mathbf{x}_0)$. The proof is now complete. □

The consequence of Theorem 2.8 is immediate: Essentially, any other (semi)norm defined on $\mathcal{H}$ (which may or may not be compatible with the topology of $\mathcal{H}$) can *not* be admissible. Obviously if f is admissible and positively homogeneous with degree $d > 0$ (*i.e.*, $f(\lambda\mathbf{x}) = \lambda^d \cdot f(\mathbf{x})$) then we have $f(\mathbf{x}) = \|\mathbf{x}\|_{\mathcal{H}}^d \cdot f(\mathbf{x}_0)$ for some (arbitrary) $\mathbf{x}_0 \in \operatorname{dom} f$ with unit norm.

Yu et al. (2013) also extended the results in this section to the matrix setting, although the characterization there is less complete.

2.5 Summary

Motivated by some existing results which all suggest the possibility to decompose the proximal map of a sum of functions into the composition of the proximal maps of the individual summands, we

first give a positive answer in the one dimensional space and a negative example in general. Then, we identify a simple sufficient condition that, if satisfied, will imply the desired decomposition. Furthermore, we completely characterize the function class that decomposes with respect to *all* positively homogeneous functions; it simply consists of all increasing radial functions. An unexpected connection to the characterization of the representer theorem in kernel methods is exposed. Finally, we generalize the prox-decomposition rule to polyhedral functions, under the cone invariance assumption. We recover most known decomposition results, with some new ones obtained almost effortlessly from our theory.

Chapter 3

Proximal Average Approximation

In Chapter 2 we discussed a particular decomposition rule for computing the proximal map of a sum of functions. Unfortunately, this rule does not always apply. In this chapter we introduce a general recipe that is based on the golden principle: We approximate "complicated" functions with more "friendly" ones. Traditionally, the nonsmooth regularizers are usually approximated by *smooth* functions. We re-examine this powerful methodology and point out a *nonsmooth* approximation which simply pretends the linearity of the proximal map. The new approximation is justified using a recent convex analysis tool—proximal average, and yields a different proximal gradient algorithm that is *strictly* better than the one based on smoothing, without incurring any extra overhead. Numerical experiments conducted on two important applications, overlapping group LASSO (*cf.* Example 1.7) and graph-guided fused LASSO (*cf.* Example 1.9), corroborate the theoretical claims.

The results in this chapter appeared in Yu (2013a).

3.1 Introduction

In many scientific areas, an important methodology that has withstood the test of time is the approximation of "complicated" functions by those that are easier to handle. For instance, Taylor's expansion in calculus (Rudin 1976), essentially a polynomial approximation of differentiable functions, has fundamentally changed analysis, and mathematics more broadly. Approximations are also ubiquitous in optimization algorithms, *e.g.* various gradient-type algorithms approximate the objective function with a quadratic upper bound. In some (if not all) cases, there are multiple ways to make the approximation, and one usually has this freedom of choice. It is perhaps not hard to convince oneself that there is no approximation that would work best in all scenarios. And one would probably also agree that a specific form of approximation should be favored if it well suits our *ultimate* goal. Despite of all these common-sense, in optimization algorithms, *smooth* approximations are still dominating, bypassing some recent advances on optimizing nonsmooth functions, see *e.g.* the last three algorithms we reviewed in Chapter 1. Part of the reason, we believe, is the lack of new technical tools.

We consider the composite minimization problem (1.1) where the objective consists of a smooth loss function and a sum of *nonsmooth* functions. Such problems have received increasing attention due to the arise of *structured sparsity* (Bach et al. 2012), notably the overlapping group LASSO (Zhao et al. 2009), the graph-guided fused LASSO (Hoefling 2010; Kim and Xing 2009) and some others. These structured regularizers, although greatly enhance our modeling capability, introduce significant new computational challenges as well. Popular gradient-type algorithms dealing with such composite problems include the generic subgradient method (Shor 1985), (accelerated) proximal gradient (APG) (Beck and Teboulle 2009; Nesterov 2013), and the smoothed accelerated proximal gradient (S-APG) of Nesterov (2005). The subgradient method is applicable to any nonsmooth function, although the convergence rate is rather slow. APG, being a recent advance, can handle *simple* functions, see *e.g.* Combettes and Pesquet (2011), Bach et al. (2011, §3.3), Parikh and Boyd (2013, §6), but for more complicated structured regularizers, an inner iterative procedure is needed, resulting in an overall convergence rate that could be as slow as the subgradient method (Villa et al. 2013). Lastly, S-APG simply runs APG on a smooth approximation of the original objective, resulting in a much improved convergence rate.

Our work is inspired by the recent advance on nonsmooth optimization, such as Beck and Teboulle (2009); Duchi et al. (2010); Nesterov (2013); Xiao (2010), of which the building block is the proximal map of the nonsmooth function. This proximal map is available in closed-form for simple functions but can be quite expensive for more complicated functions such as a *sum* of nonsmooth functions we consider here. A key observation we make is that oftentimes the proximal map for each individual summand can be easily computed, therefore a bold idea is to simply use the sum of proximal maps, pretending that the proximal map is a linear operator. Somewhat surprisingly, this naive idea, when combined with APG, results in a new proximal algorithm that is *strictly* better than S-APG, while keeping per-step complexity unchanged. We justify our method via a new tool from convex analysis—the proximal average (Bauschke et al. 2008). In essence, instead of smoothing the nonsmooth function, we use a nonsmooth approximation whose proximal map is cheap to evaluate, after all this is all we need to run APG.

We formally state our problem in Section 3.2, along with the proposed algorithm. After recalling the relevant tools from convex analysis in Section 3.3 we provide the theoretical justification of our method in Section 3.4. Related works are discussed in Section 3.5 and refinements are presented in Section 3.6. We test the proposed algorithm in Section 3.7 and conclude in Section 3.8.

3.2 Problem Formulation

We are interested in solving the following composite minimization problem:

$$\inf_{\mathbf{w}\in\mathcal{H}} \ell(\mathbf{w}) + \bar{f}(\mathbf{w}), \quad \text{where} \quad \bar{f}(\mathbf{w}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{w}). \tag{3.1}$$

Here ℓ is convex with L_0-Lipschitz continuous gradient[1] *w.r.t.* the Hilbertian norm $\|\cdot\|_{\mathcal{H}}$, and $\alpha_k \geq 0, \sum_k \alpha_k = 1$. The usual regularization constant that balances the two terms in (3.1) is absorbed into the loss ℓ. For the functions f_k, we make the following assumption.

Assumption 3.1. *Each f_k is convex and M_k-Lipschitz continuous* w.r.t. *the Hilbertian norm $\|\cdot\|_{\mathcal{H}}$.*

The abbreviation $\overline{M^2} = \sum_{k=1}^{K} \alpha_k M_k^2$ is adopted throughout.

We are interested in the general case where the functions f_k need not be differentiable. As mentioned in the introduction, a generic scheme that solves (3.1) is the subgradient method (Shor 1985), of which each step requires merely an arbitrary subgradient of the objective. With a suitable step size, the subgradient method converges[2] in at most $O(1/\epsilon^2)$ steps where $\epsilon > 0$ is the desired accuracy, see Section 1.2 for details. Although being general, the subgradient method is exceedingly slow, making it unsuitable for many practical applications.

Another recent algorithm for solving (3.1) is the (accelerated) proximal gradient (APG) (Beck and Teboulle 2009; Combettes and Wajs 2005; Nesterov 2013), of which each iteration needs to compute the proximal map of the nonsmooth part $\bar{f}$ in (3.1):

$$\mathsf{P}_{\bar{f}}^{1/L_0}(\mathbf{w}) = \operatorname*{argmin}_{\mathbf{z}} \tfrac{L_0}{2}\|\mathbf{w}-\mathbf{z}\|^2 + \bar{f}(\mathbf{z}).$$

Recall that L_0 is the Lipschitz constant of the gradient of the smooth part ℓ in (3.1). Provided that the proximal map can be computed easily, it can be shown that APG converges within $O(1/\sqrt{\epsilon})$ steps, significantly better than the subgradient method, see Section 1.3 for details. For some simple functions, the proximal map indeed is available in closed-form, see Combettes and Pesquet (2011), Bach et al. (2011, §3.3), Parikh and Boyd (2013, §6) for nice summaries. However, for more complicated functions such as the one we consider here, the proximal map itself is expensive to compute and an inner iterative subroutine is required. Somewhat disappointingly, recent analysis has shown that such a two-loop procedure can be as slow as the subgradient method (Villa et al. 2013).

Yet another approach, popularized by Nesterov (2005), is to approximate each nonsmooth component f_k with a smooth function and then run APG. By carefully balancing the approximation and the convergence requirement of APG, the smoothed accelerated proximal gradient (S-APG) proposed by Nesterov (2005) converges in at most $O(\sqrt{1/\epsilon^2 + 1/\epsilon})$ steps, again much better than the subgradient method. However, the downside is that smoothing always increases the Lipschitz constant. The main point of this chapter is to further improve S-APG, in perhaps a surprisingly simple way.

The key assumption that we will exploit is the following:

Assumption 3.2. *Each proximal map $\mathsf{P}_{f_k}^{\eta}$ can be computed easily for any $\eta > 0$.*

[1] Namely $\|\nabla\ell(\mathbf{x}) - \nabla\ell(\mathbf{y})\|_{\mathcal{H}} \leq L_0 \cdot \|\mathbf{x}-\mathbf{y}\|_{\mathcal{H}}$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{H}$.

[2] We satisfy ourselves with convergence in terms of function values, although with additional assumptions/efforts it is possible to argue for convergence in terms of the iterates.

Algorithm 4 PA-APG.

Initialize $\mathbf{w}_0 = \mathbf{u}_1$, $\eta = \min\{1/L_0, 2\epsilon/\overline{M^2}\}$, $\gamma_1 = 1$.
for $t = 1, 2, \ldots$ **do**
  $\mathbf{z}_t = \mathbf{u}_t - \eta\nabla\ell(\mathbf{u}_t)$,
  $\mathbf{w}_t = \sum_k \alpha_k \cdot \mathsf{P}^{\eta}_{f_k}(\mathbf{z}_t)$,
  $\gamma_{t+1} = \frac{1+\sqrt{1+4\gamma_t^2}}{2}$,
  $\mathbf{u}_{t+1} = \mathbf{w}_t + \frac{\gamma_t - 1}{\gamma_{t+1}}(\mathbf{w}_t - \mathbf{w}_{t-1})$.
end for

Algorithm 5 PA-PG.

Initialize $\mathbf{w}_0$, $\eta = \min\{1/L_0, 2\epsilon/\overline{M^2}\}$.
for $t = 1, 2, \ldots$ **do**
  $\mathbf{z}_t = \mathbf{w}_{t-1} - \eta\nabla\ell(\mathbf{w}_{t-1})$,
  $\mathbf{w}_t = \sum_k \alpha_k \cdot \mathsf{P}^{\eta}_{f_k}(\mathbf{z}_t)$.
end for

We prefer to leave the exact meaning of "easily" unspecified, but roughly speaking, the proximal map should be no more expensive than computing the gradient of the smooth part ℓ so that it does not become the bottleneck. Both Assumption 3.1 and Assumption 3.2 are satisfied in many important applications (examples will follow). As it will also become clear later, these assumptions are exactly those needed by S-APG.

Unfortunately, in general, there is no known *efficient* way that reduces the proximal map of the average $\bar{f}$ to the proximal maps of its individual components f_k, therefore the fast schemes PG or APG are not readily applicable. The main difficulty, of course, is due to the nonlinearity of the proximal map P^{η}_{f}, when treated as a map on the function f. Despite of this fact, we will "naively" pretend that the proximal map is linear and use[3]

$$\mathsf{P}^{\eta}_{\bar{f}} \overset{?}{\approx} \sum_{k=1}^{K} \alpha_k \mathsf{P}^{\eta}_{f_k}. \tag{3.2}$$

Under this approximation, the fast schemes PG or APG can be applied. We give one particular realization (PA-APG) in Algorithm 4 based on the FISTA of Beck and Teboulle (2009). A simpler, though slower, version (PA-PG) based on the ISTA of Beck and Teboulle (2009) is also provided in Algorithm 5. Clearly both algorithms are easily parallelizable if K is large. We remark that any other variant of APG or PG, *e.g.* Nesterov (2005), is equally well applicable. Of course, when $K = 1$, our algorithm reduces to the corresponding APG or PG scheme.

At this point, one might be suspicious about the usefulness of the "naive" approximation in (3.2). Before addressing this well-deserved question, let us first point out two important applications where Assumption 3.1 and Assumption 3.2 are naturally satisfied.

[3] Of course, this idea, *per se*, is not new at all, as we use linear approximations everywhere and all the time.

Example 1.7 (continuing from p. 16)**.** *Recall that in overlapping group LASSO, we let* $f_k(\mathbf{w}) = \|\mathbf{w}_{g_k}\|_{\mathcal{H}}$ *where* g_k *is a group (subset) of variables and* $\mathbf{w}_g$ *denotes the restriction of* $\mathbf{w}$ *to the variables belong to the group* g*. This group regularizer has been proven quite useful in high-dimensional statistics with the capability of selecting meaningful groups of features (Zhao et al. 2009). In the general case where the groups could overlap as needed,* $\mathsf{P}^{\eta}_{\bar{f}}$ *cannot be computed easily.*

Clearly each f_k *is convex and* 1*-Lipschitz continuous* w.r.t. $\|\cdot\|_{\mathcal{H}}$, i.e., $M_k = 1$ *in Assumption 3.1. Moreover, the proximal map* $\mathsf{P}^{\eta}_{f_k}$ *is simply a re-scaling of the variables in group* g_k*, that is*

$$[\mathsf{P}^{\eta}_{f_k}(\mathbf{w})]_j = \begin{cases} w_j, & j \notin g_k \\ \left(1 - \eta / \|\mathbf{w}_{g_k}\|_{\mathcal{H}}\right)_+ w_j, & j \in g_k \end{cases}, \tag{3.3}$$

where recall that $(\lambda)_+ = \max\{\lambda, 0\}$*. Therefore, both of our assumptions are met.*

Example 1.9 (continuing from p. 18)**.** *Recall that this example is an enhanced version of the fused LASSO (Tibshirani et al. 2005), with some graph structure exploited to improve feature selection in biostatistic applications (Kim and Xing 2009). Specifically, given some graph whose nodes correspond to the feature variables, we let* $f_{ij}(\mathbf{w}) = |w_i - w_j|$ *for every edge* $\{i, j\} \in E$*. For a general graph, the proximal map of the regularizer* $\bar{f} = \sum_{\{i,j\}\in E} \alpha_{ij} f_{ij}$ *with* $\alpha_{ij} \geq 0, \sum_{\{i,j\}\in E} \alpha_{ij} = 1$ *is not easily computable.*

Similar as above, each f_{ij} *is* 1*-Lipschitz* w.r.t. *the Hilbertian norm. Moreover, the proximal map* $\mathsf{P}^{\eta}_{f_{ij}}$ *is easy to compute:*

$$[\mathsf{P}^{\eta}_{f_{ij}}(\mathbf{w})]_s = \begin{cases} w_s, & s \notin \{i, j\} \\ w_s - \operatorname{sign}(w_i - w_j) \min\{\eta, |w_i - w_j|/2\}, & s \in \{i, j\} \end{cases}. \tag{3.4}$$

Again, both our assumptions are satisfied.

Note that in both examples we could have incorporated weights into the component functions f_k or f_{ij}, which amounts to changing α_k or α_{ij} accordingly. We also remark that there are other applications that fall into our consideration, for instance, SVM in Example 1.2 if we swap the role of loss and regularizer. For illustration purposes we shall contend ourselves with the above two examples. More conveniently, both examples have been tried with S-APG by Chen et al. (2012), thus constitute a natural benchmark for our new algorithm.

3.3 Technical Tools

To justify our new algorithm, we need a few technical tools from convex analysis (Rockafellar and Wets 1998). Recall that Γ_0 denotes the set of all closed proper convex functions $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. For any $f \in \Gamma_0$, its Fenchel conjugate

$$f^*(\mathbf{z}) = \sup_{\mathbf{w}\in\mathcal{H}} \langle \mathbf{w}, \mathbf{z} \rangle - f(\mathbf{w})$$

also belongs to Γ_0. Moreover, $(f^*)^* = f$. For convenience, throughout we let $\mathsf{q} = \frac{1}{2}\|\cdot\|^2_{\mathcal{H}}$ (q for "quadratic"). Note that q is the only function which coincides with its Fenchel conjugate. Another

convention that we borrow from convex analysis is to write $(f\eta)(\mathbf{w}) = \eta f(\eta^{-1}\mathbf{w})$ for $\eta > 0$, while $(\eta f)(\mathbf{w}) = \eta \cdot f(\mathbf{w})$ as usual. We easily verify that $(\eta f)^* = f^*\eta$ and also $(f\eta)^* = \eta f^*$, *i.e.*, "left" and "right" (positive) scalar multiplications interchange after taking the Fenchel conjugation.

For any $f \in \Gamma_0$, we define its Moreau envelop, with parameter $\eta > 0$, as

$$\mathsf{M}_f^\eta(\mathbf{z}) = \min_{\mathbf{x}\in\mathcal{H}} \tfrac{1}{2\eta} \|\mathbf{z}-\mathbf{x}\|_{\mathcal{H}}^2 + f(\mathbf{x}), \tag{3.5}$$

and correspondingly the proximal map

$$\mathsf{P}_f^\eta(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{H}} \tfrac{1}{2\eta} \|\mathbf{z}-\mathbf{x}\|_{\mathcal{H}}^2 + f(\mathbf{x}). \tag{3.6}$$

Since $f \in \Gamma_0$ and q is strongly convex, the proximal map is well-defined and single-valued. As mentioned before, the proximal map is the key component of many gradient algorithms such as PG or APG. In fact, Moreau (1965) originally considered only $\eta \equiv 1$. The parameter η, introduced as a means to regularize f, seems to be due to Attouch (1984). In Chapter 2 we chose to absorb η into the function f and considered a certain decomposition rule for the proximal map. Here, the parameter η is made explicit so that we can control a certain form of approximation to f. Intuitively, as $\eta \to 0$, the envelop $\mathsf{M}_f^\eta \to f$ in a pointwise manner while as $\eta \to \infty$, $\mathsf{M}_f^\eta \equiv \inf_{\mathbf{w}\in\mathcal{H}} f(\mathbf{w})$, although a rigorous justification requires some effort, see Rockafellar and Wets (1998).

We mentioned some nice properties of the Moreau envelop and the proximal map in Chapter 2. For this chapter's purpose, we document some additional properties below.

Proposition 3.1. *Let $\eta, \lambda > 0$, $f \in \Gamma_0$, and Id be the identity map, then*

i). $\mathsf{M}_f^\eta \in \Gamma_0$ *and* $(\mathsf{M}_f^\eta)^* = f^* + \eta\mathsf{q}$;

ii). $\mathsf{M}_f^\eta \le f$, $\inf_{\mathbf{z}} \mathsf{M}_f^\eta(\mathbf{z}) = \inf_{\mathbf{z}} f(\mathbf{z})$, *and* $\operatorname{argmin}_{\mathbf{z}} \mathsf{M}_f^\eta(\mathbf{z}) = \operatorname{argmin}_{\mathbf{z}} f(\mathbf{z})$;

iii). M_f^η *is differentiable with* $\nabla \mathsf{M}_f^\eta = \frac{1}{\eta}(\mathsf{Id} - \mathsf{P}_f^\eta)$;

iv). $\mathsf{M}_{\lambda f}^\eta = \lambda \mathsf{M}_f^{\lambda\eta}$ *and* $\mathsf{P}_{\lambda f}^\eta = \mathsf{P}_f^{\lambda\eta} = (\mathsf{P}_{f\lambda^{-1}}^\eta)\lambda$;

v). $\mathsf{M}_{\mathsf{M}_f^\eta}^\lambda = \mathsf{M}_f^{\lambda+\eta}$ *and* $\mathsf{P}_{\mathsf{M}_f^\eta}^\lambda = \frac{\eta}{\lambda+\eta}\mathsf{Id} + \frac{\lambda}{\lambda+\eta}\mathsf{P}_f^{\lambda+\eta}$;

vi). $\eta\mathsf{M}_f^\eta + (\mathsf{M}_{f^*}^{1/\eta})\eta = \mathsf{q}$ *and* $\mathsf{P}_f^\eta + (\mathsf{P}_{f^*}^{1/\eta})\eta = \mathsf{Id}$.

i) is the well-known duality between infimal convolution and summation. ii), albeit being trivial, is the driving force behind the proximal point algorithm (Martinet 1970; Rockafellar 1976). iii) justifies the "niceness" of the Moreau envelop and connects it to the proximal map in a more convenient way. iv) and v) follow from simple algebra. And lastly vi), known as Moreau's identity, plays an important role in the early development of convex analysis. We remind that $(\mathsf{M}_f^\eta)^*$ in general is different from $\mathsf{M}_{f^*}^\eta$.

Let us elaborate on the usefulness of the Moreau envelop in optimization. Suppose we want to minimize some function $f \in \Gamma_0$, possibly nonsmooth or very ill-conditioned. Thanks to ii) in

Proposition 3.1, we can w.l.o.g. consider instead the envelop M_f^η, which is always a differentiable function. Even better, we get to choose any parameter $\eta > 0$. Applying the usual gradient descent to M_f^η we get the update rule

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \mathsf{M}_f^\eta(\mathbf{w}_t) = \mathsf{P}_f^\eta(\mathbf{w}_t),$$

where the second equality follows from iv) in Proposition 3.1. This is exactly the proximal point algorithm of Martinet (1970), see also Rockafellar (1976), who further considered varying η and error tolerance. Of course, the caveat is that computing the proximal map P_f^η might be as hard as minimizing f directly. An important exception is when f is a quadratic function, whose Moreau envelop is again quadratic but much better conditioned (depending on how big η is). The proximal gradient algorithm we saw in Section 1.3 was proposed by Fukushima and Mine (1981) as a linearization, hence also generalization, of the proximal point algorithm. The similarity is evident. The idea to use the Moreau envelop as a means to regularize "bad" functions has proven very fruitful, and has been used in many fields, sometimes even without noticing the connection. See Attouch (1984); Rockafellar and Wets (1998) for more discussions.

Fix $\eta > 0$. Let $\mathsf{SC}_\eta \subseteq \Gamma_0$ denote the class of η-strongly convex functions, that is, functions f such that $f - \eta\mathsf{q}$ is convex. Similarly, let $\mathsf{SS}_\eta \subseteq \Gamma_0$ denote the class of finite-valued functions whose gradient is η-Lipschitz continuous *w.r.t.* the norm $\|\cdot\|_{\mathcal{H}}$. A well-known duality between strong convexity and smoothness is that for $f \in \Gamma_0$, $f \in \mathsf{SC}_\eta$ if and only if $f^* \in \mathsf{SS}_{1/\eta}$, *cf.* Zălinescu (2002, Corollary 3.5.11). We have used this result to present a much cleaner view of RDA in Section 1.5. The next result, also based on this duality, turns out to be critical.

Proposition 3.2. *Fix $\eta > 0$. The Moreau envelop map $\mathsf{M}^\eta : \Gamma_0 \to \mathsf{SS}_{1/\eta}$ that sends $f \in \Gamma_0$ to M_f^η is bijective, increasing, and concave on any convex subset of Γ_0 (under the pointwise order)*[4].

Proof. Fix $f, g \in \Gamma_0$. First note that the Fenchel conjugation enjoys (and is characterized by!) the order reversing property:

$$f \geq g \iff f^* \leq g^*.$$

Since $(\mathsf{M}_f^\eta)^* = f^* + \eta\mathsf{q} \in \mathsf{SC}_\eta$ we have $\mathsf{M}_f^\eta \in \mathsf{SS}_{1/\eta}$. On the other hand, let $h \in \mathsf{SS}_{1/\eta}$. Then $g = h^* - \eta\mathsf{q} \in \Gamma_0$, hence $h^* = g + \eta\mathsf{q}$ and $h = (g + \eta\mathsf{q})^* = \mathsf{M}_{g^*}^\eta$. Therefore M^η is onto.

It should be clear that $\mathsf{M}^\eta : \Gamma_0 \to \mathsf{SS}_{1/\eta}$ is increasing *w.r.t.* the pointwise order, *i.e.*, $f \geq g \implies \mathsf{M}_f^\eta \geq \mathsf{M}_g^\eta$. On the other hand, $\mathsf{M}_f^\eta \geq \mathsf{M}_g^\eta \implies (\mathsf{M}_f^\eta)^* \leq (\mathsf{M}_g^\eta)^*$, which, by i) in Proposition 3.1, means $f^* + \eta\mathsf{q} \leq g^* + \eta\mathsf{q} \implies f^* \leq g^* \implies f = f^{**} \geq g^{**} = g$. Hence M^η is an injection.

Let $\alpha \in]0, 1[$, then

$$\begin{aligned}
\mathsf{M}_{\alpha f + (1-\alpha) g}^\eta(\mathbf{z}) &= \min_{\mathbf{x}} \tfrac{1}{2\eta} \|\mathbf{z} - \mathbf{x}\|_{\mathcal{H}}^2 + \alpha f(\mathbf{x}) + (1-\alpha) g(\mathbf{x}) \\
&= \min_{\mathbf{x}} \tfrac{\alpha}{2\eta} \|\mathbf{z} - \mathbf{x}\|_{\mathcal{H}}^2 + \alpha f(\mathbf{x}) + \tfrac{1-\alpha}{2\eta} \|\mathbf{z} - \mathbf{x}\|_{\mathcal{H}}^2 + (1-\alpha) g(\mathbf{x})
\end{aligned}$$

[4]The reason to restrict to convex subsets is that Γ_0 itself is not convex: $\frac{f+g}{2}$ might not be proper even when f and g both are.

$$\geq \min_{\mathbf{x}} \tfrac{\alpha}{2\eta} \|\mathbf{z}-\mathbf{x}\|_{\mathcal{H}}^2 + \alpha f(y) + \min_{\mathbf{x}} \tfrac{1-\alpha}{2\eta} \|\mathbf{z}-\mathbf{x}\|_{\mathcal{H}}^2 + (1-\alpha) g(y)$$
$$= \alpha \mathsf{M}_f^\eta(\mathbf{z}) + (1-\alpha)\mathsf{M}_g^\eta(\mathbf{z}),$$

verifying the concavity of M^η. □

It is clear that $\mathsf{SS}_{1/\eta}$ is a *convex*[5] subset of Γ_0, which motivates the definition of the proximal average—the key object to us. Fix constants $\alpha_k \geq 0$ with $\sum_{k=1}^K \alpha_k = 1$. Recall that $\bar{f} = \sum_k \alpha_k f_k$ with each $f_k \in \Gamma_0$, *i.e.* $\bar{f}$ is the convex combination of the component functions $\{f_k\}$ under the weight $\{\alpha_k\}$.

Definition 3.1 (Proximal Average, Bauschke et al. (2008); Moreau (1965))**.** *Denote* $\mathbf{f} = (f_1, \ldots, f_K)$ *and* $\mathbf{f}^* = (f_1^*, \ldots, f_K^*)$*. The proximal average* $\mathsf{A}^\eta_{\mathbf{f},\boldsymbol{\alpha}}$*, or simply* A^η *when the component functions and weights are clear from context, is the unique function* $h \in \Gamma_0$ *such that* $\mathsf{M}_h^\eta = \sum_{k=1}^K \alpha_k \mathsf{M}_{f_k}^\eta$.

Indeed, the existence of the proximal average follows from the surjectivity of M^η while the uniqueness follows from the injectivity of M^η, both proven in Proposition 3.2. The main property of the proximal average, as seen from its definition, is that its Moreau envelop is the convex combination of the Moreau envelops of the component functions. By iii) of Proposition 3.1 we immediately obtain

$$\mathsf{P}^\eta_{\mathsf{A}^\eta} = \sum_{k=1}^K \alpha_k \mathsf{P}^\eta_{f_k}. \tag{3.7}$$

Recall that the right-hand side is exactly the approximation we employed in Section 3.2.

Interestingly, using the properties we summarized in Proposition 3.1, we can show that the Fenchel conjugate of the proximal average, denoted as $(\mathsf{A}^\eta)^*$, enjoys a similar property (Bauschke et al. 2008):

$$\left[\mathsf{M}^{1/\eta}_{(\mathsf{A}^\eta)^*}\right]\eta = \mathsf{q} - \eta \mathsf{M}^\eta_{\mathsf{A}^\eta} = \mathsf{q} - \eta \sum_{k=1}^K \alpha_k \mathsf{M}^\eta_{f_k} = \sum_{k=1}^K \alpha_k (\mathsf{q} - \eta \mathsf{M}^\eta_{f_k})$$
$$= \sum_{k=1}^K \alpha_k [(\mathsf{M}^{1/\eta}_{f_k^*})\eta] = \left[\sum_{k=1}^K \alpha_k \mathsf{M}^{1/\eta}_{f_k^*}\right]\eta,$$

that is, $\mathsf{M}^{1/\eta}_{(\mathsf{A}^\eta_{\mathbf{f},\boldsymbol{\alpha}})^*} = \sum_{k=1}^K \alpha_k \mathsf{M}^{1/\eta}_{f_k^*} = \mathsf{M}^{1/\eta}_{\mathsf{A}^{1/\eta}_{\mathbf{f}^*,\boldsymbol{\alpha}}}$, therefore by the injective property established in Proposition 3.2:

$$(\mathsf{A}^\eta_{\mathbf{f},\boldsymbol{\alpha}})^* = \mathsf{A}^{1/\eta}_{\mathbf{f}^*,\boldsymbol{\alpha}}. \tag{3.8}$$

From its definition it is also possible to derive an explicit formula for the proximal average (although for our purpose only the existence is needed):

$$\mathsf{A}^\eta_{\mathbf{f},\boldsymbol{\alpha}} = \left(\left(\sum_{k=1}^K \alpha_k \mathsf{M}^\eta_{f_k}\right)^* - \eta \mathsf{q}\right)^* = \left(\sum_{k=1}^K \alpha_k \mathsf{M}^{1/\eta}_{f_k^*}\right)^* - \mathsf{q}\eta, \tag{3.9}$$

[5] In contrast, SC_η is not convex: the convex combination of two proper functions need not be proper.

where the second equality is obtained by conjugating (3.8) and applying the first equality to the conjugate. By the concavity and monotonicity of M^η, we have the inequality

$$\mathsf{M}^\eta_{\bar{f}} \geq \sum_{k=1}^K \alpha_k \mathsf{M}^\eta_{f_k} = \mathsf{M}^\eta_{\mathsf{A}^\eta} \iff \bar{f} \geq \mathsf{A}^\eta. \tag{3.10}$$

It is well-known that as $\eta \to 0$, $\mathsf{M}^\eta_f \to f$ pointwise (Rockafellar and Wets 1998), which, under the Lipschitz assumption, can be strengthened to uniform convergence:

Proposition 3.3. *Under Assumption 3.1 we have* $0 \leq \bar{f} - \mathsf{M}^\eta_{\mathsf{A}^\eta} \leq \frac{\eta \overline{M^2}}{2}$, *where recall that* $\bar{f} = \sum_k \alpha_k f_k$, $\overline{M^2} = \sum_{k=1}^K \alpha_k M_k^2$ *and* M_k *is the Lipschitz constant of* f_k.

Proof. First observe that by the definition of the proximal average

$$\bar{f} - \mathsf{M}^\eta_{\mathsf{A}^\eta} = \sum_k \alpha_k (f_k - \mathsf{M}^\eta_{f_k}) \geq 0,$$

since $f \geq \mathsf{M}^\eta_f$ for any $f \in \Gamma_0$. On the other hand

$$\begin{aligned}
\sup_{\mathbf{z}} f_k(\mathbf{z}) - \mathsf{M}^\eta_{f_k}(\mathbf{z}) &= \sup_{\mathbf{z}} f_k(\mathbf{z}) - \min_{\mathbf{x}} \tfrac{1}{2\eta} \|\mathbf{z} - \mathbf{x}\|^2_{\mathcal{H}} + f_k(\mathbf{x}) \\
&= \sup_{\mathbf{z},\mathbf{x}} f_k(\mathbf{z}) - f_k(\mathbf{x}) - \tfrac{1}{2\eta} \|\mathbf{z} - \mathbf{x}\|^2_{\mathcal{H}} \\
&\leq \sup_{\mathbf{z},\mathbf{x}} M_k \|\mathbf{z} - \mathbf{x}\|_{\mathcal{H}} - \tfrac{1}{2\eta} \|\mathbf{z} - \mathbf{x}\|^2_{\mathcal{H}} \\
&\leq \tfrac{\eta M_k^2}{2},
\end{aligned}$$

where the first inequality is due to the Lipschitz assumption on f_k. Therefore

$$\sup_{\mathbf{z}} \bar{f}(\mathbf{z}) - \mathsf{M}^\eta_{\mathsf{A}^\eta}(\mathbf{z}) \leq \sum_k \alpha_k \left[\sup_{\mathbf{z}} f_k(\mathbf{z}) - \mathsf{M}^\eta_{f_k}(\mathbf{z}) \right] \leq \tfrac{\eta \overline{M^2}}{2}.$$

□

For the proximal average, Bauschke et al. (2008) showed that $\mathsf{A}^\eta \to \bar{f}$ pointwise, which again can be strengthened to uniform convergence.

Proposition 3.4. *Under Assumption 3.1 we have* $0 \leq \bar{f} - \mathsf{A}^\eta \leq \frac{\eta \overline{M^2}}{2}$.

Proof. The claim follows immediately from (3.10) and Proposition 3.3 since $\mathsf{A}^\eta \geq \mathsf{M}^\eta_{\mathsf{A}^\eta}$. □

As it turns out, S-APG approximates the nonsmooth function $\bar{f}$ with the smooth function $\mathsf{M}^\eta_{\mathsf{A}^\eta}$ while our algorithm operates on the *nonsmooth* approximation A^η (note that it can be shown that A^η is smooth iff some component f_i is smooth). By (3.10) and ii) in Proposition 3.1 we have

$$\mathsf{M}^\eta_{\mathsf{A}^\eta} \leq \mathsf{A}^\eta \leq \bar{f}, \tag{3.11}$$

meaning that the proximal average A^η is a better under-approximation of $\bar{f}$ than $\mathsf{M}^\eta_{\mathsf{A}^\eta}$.

Let us compare the proximal average A^η with the smooth approximation $\mathsf{M}^\eta_{\mathsf{A}^\eta}$ on a 1-D example.

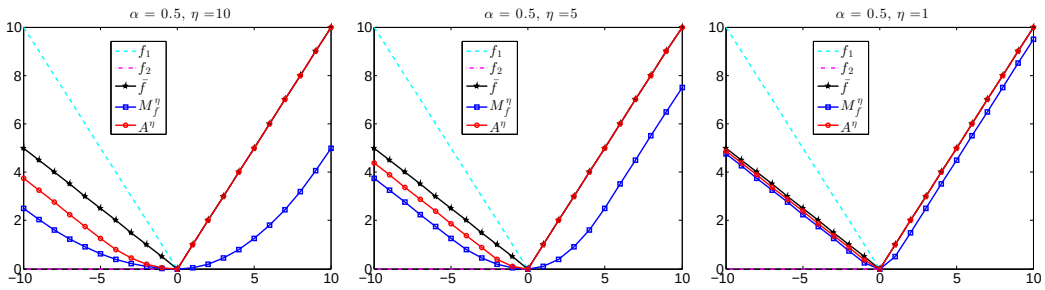


Figure 3.1: Comparison of the Moreau envelop and the proximal average. See Example 3.1 for context.

Example 3.1. *Let* $f_1(x) = |x|, f_2(x) = \max\{x, 0\}$. *Clearly both are* 1*-Lipschitz. Moreover,* $\mathsf{P}^{\eta}_{f_1}(x) = \operatorname{sign}(x)(|x| - \eta)_+$, $\mathsf{P}^{\eta}_{f_2}(x) = (x - \eta)_+ + x - (x)_+$,

$$\mathsf{M}^{\eta}_{f_1}(x) = \begin{cases} \frac{x^2}{2\eta}, & |x| \le \eta \\ |x| - \eta/2, & \textit{otherwise} \end{cases}, \textit{ and } \mathsf{M}^{\eta}_{f_2}(x) = \begin{cases} 0, & x \le 0 \\ \frac{x^2}{2\eta}, & 0 \le x \le \eta \\ x - \eta/2, & \textit{otherwise} \end{cases}.$$

Finally, using (3.9) *we obtain (with* $\alpha_1 = \alpha, \alpha_2 = 1 - \alpha$*)*

$$\mathsf{A}^{\eta}(x) = \begin{cases} x, & x \ge 0 \\ \frac{\alpha}{1-\alpha}\frac{x^2}{2\eta}, & (\alpha - 1)\eta \le x \le 0 \\ -\alpha x - (1-\alpha)\frac{\alpha\eta}{2}, & x \le (\alpha - 1)\eta \end{cases}.$$

Figure 3.1 depicts the case $\alpha = 0.5$ *with different values of the smoothing parameter* η*. As predicted* $\mathsf{M}^{\eta}_{\mathsf{A}^{\eta}} \le \mathsf{A}^{\eta} \le \bar{f}$*. Observe that the proximal average* A^{η} *remains nondifferentiable at 0 while* $\mathsf{M}^{\eta}_{\mathsf{A}^{\eta}}$ *is smooth everywhere. For* $x \ge 0$*,* $f_1 = f_2 = \bar{f} = \mathsf{A}^{\eta}$ *(the red circled line), thus the proximal average* A^{η} *is a strictly tighter approximation than smoothing. When* η *is small (right panel),* $\bar{f} \approx \mathsf{M}^{\eta}_{\mathsf{A}^{\eta}} \approx \mathsf{A}^{\eta}$*.*

3.4 Theoretical Justification

Given our development in the previous section, it is now clear that our proposed Algorithm 4 and Algorithm 5 aim at solving the approximation

$$\min_{\mathbf{w}} \ell(\mathbf{w}) + \mathsf{A}^{\eta}(\mathbf{w}). \tag{3.12}$$

The next important piece is to show how a careful choice of η would lead to a strictly better convergence rate than S-APG.

Recall that using APG to solve (3.12) requires computing the following proximal map in each iteration:

$$\mathsf{P}^{1/L_0}_{\mathsf{A}^{\eta}}(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{x}} \tfrac{L_0}{2}\|\mathbf{z} - \mathbf{x}\|^2_{\mathcal{H}} + \mathsf{A}^{\eta}(\mathbf{x}),$$

which, unfortunately, is not yet amenable to efficient computation, due to the mismatch of the constants $1/L_0$ and η (recall that in the decomposition (3.7) the superscript and subscript must both

be η). In general, there is no known explicit formula that would reduce P_f^{1/L_0} to P_f^{η} for different positive constants L_0 and η (Bauschke and Combettes 2011, p. 338), see also iv) in Proposition 3.1. Our fix is almost trivial: If necessary, we use a bigger Lipschitz constant $L_0 = 1/\eta$ so that we can compute the proximal map easily. This is indeed legitimate since L_0-Lipschitz implies L-Lipschitz for any $L \geq L_0$. Said differently, all we need is to tune down the step size a little bit in APG. We state formally the convergence property of our algorithm in the next theorem.

Theorem 3.1. *Fix the desired accuracy $\epsilon > 0$. If ℓ is finite-valued and satisfies Assumption 1.2, $f_k, k = 1, \ldots, K$, satisfy Assumption 3.1, and $\eta = \min\{1/L_0, 2\epsilon/\overline{M^2}\}$, then after at most $\sqrt{\frac{2}{\eta\epsilon}}\|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}$ steps, the output of Algorithm 4, say $\tilde{\mathbf{w}}$, satisfies*

$$\forall \mathbf{w}, \quad \ell(\tilde{\mathbf{w}}) + \bar{f}(\tilde{\mathbf{w}}) \leq \ell(\mathbf{w}) + \bar{f}(\mathbf{w}) + 2\epsilon.$$

The same guarantee holds for Algorithm 5 after at most $\frac{1}{2\eta\epsilon}\|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}^2$ steps.

Proof. Clearly, under our choice of η, the gradient of ℓ is $1/\eta$-Lipschitz continuous (since $1/\eta \geq L_0$). According to Theorem 1.3, after at most $\sqrt{\frac{2}{\eta\epsilon}}\|\mathbf{w}_0 - \mathbf{w}\|_{\mathcal{H}}$ steps the output of Algorithm 4, say $\tilde{\mathbf{w}}$, satisfies[6]

$$\ell(\tilde{\mathbf{w}}) + \mathsf{A}^{\eta}(\tilde{\mathbf{w}}) \leq \ell(\mathbf{w}) + \mathsf{A}^{\eta}(\mathbf{w}) + \epsilon. \tag{3.13}$$

Then by Proposition 3.4

$$\begin{aligned}[\ell(\tilde{\mathbf{w}}) + \bar{f}(\tilde{\mathbf{w}})] - [\ell(\mathbf{w}) + \bar{f}(\mathbf{w})] &= [\ell(\tilde{\mathbf{w}}) + \mathsf{A}^{\eta}(\tilde{\mathbf{w}})] - [\ell(\mathbf{w}) + \mathsf{A}^{\eta}(\mathbf{w})] \\ &\quad + [\bar{f}(\tilde{\mathbf{w}}) - \mathsf{A}^{\eta}(\tilde{\mathbf{w}})] - [\bar{f}(\mathbf{w}) - \mathsf{A}^{\eta}(\mathbf{w})] \\ &\leq \epsilon + \epsilon + 0 = 2\epsilon.\end{aligned}$$

The proof for Algorithm 5 is similar. □

Note that if we could reduce $\mathsf{P}_{\mathsf{A}^{\eta}}^{1/L_0}$ efficiently to $\mathsf{P}_{\mathsf{A}^{\eta}}^{\eta}$, we would end up with the optimal (overall) rate $O(\sqrt{1/\epsilon})$, even though we approximate the nonsmooth function $\bar{f}$ by the proximal average A^{η}. In other words, approximation itself does not lead to an inferior rate. It is our incapability to (efficiently) relate proximal maps that leads to the sacrifice in convergence rate. We will better illustrate this point through concrete examples in Section 3.6.

3.5 Comparing to Existing Approaches

To ease our discussion with related works, let us first point out a fact that is not always explicitly recognized, that is, S-APG essentially relies on approximating the nonsmooth function $\bar{f}$ with $\mathsf{M}_{\mathsf{A}^{\eta}}^{\eta}$.

[6] Finally, it is time to explain our obsession, besides pursuing the ultimate generality of course, in Chapter 1 to state results *w.r.t.* an arbitrary $\mathbf{w}$, instead of a minimizer $\mathbf{w}^{\star}$: Had we done that, we could only claim Theorem 3.1 after $\sqrt{\frac{2}{\eta\epsilon}}\|\mathbf{w}_0 - \mathbf{w}_{\eta}^{\star}\|_{\mathcal{H}}$ steps, yielding another unnecessary, albeit implicit, dependence on η. I am grateful to an anonymous NIPS reviewer who brought this issue into my attention.

Indeed, consider first the case $K = 1$. The smoothing idea introduced in Nesterov (2005) purports the superficial max-structure assumption, that is, $f(\mathbf{z}) = \max_{\mathbf{x}\in C} \langle \mathbf{x}, \mathbf{z}\rangle - h(\mathbf{x})$ where C is some closed and bounded convex set and $h \in \Gamma_0$. As it is readily verified from the definition, $f \in \Gamma_0$ is M-Lipschitz continuous (*w.r.t.* the norm $\|\cdot\|$) if and only if $\operatorname{dom} f^* \subseteq \mathsf{B}_{\|\cdot\|}(\mathbf{0}, M)$, the ball centered at the origin with radius M. Thus the function $f \in \Gamma_0$ admits Nesterov's max-structure if and only if it is Lipschitz continuous, *i.e.*, satisfying our Assumption 3.1, in which case $h = f^*$ and $C = \operatorname{dom} f^*$. Nesterov (2005) proceeded to add some "distance" function d to obtain the approximation $f_\eta(\mathbf{z}) = \max_{\mathbf{x}\in C} \langle \mathbf{x}, \mathbf{z}\rangle - f^*(\mathbf{x}) - \eta\mathsf{d}(\mathbf{x})$. For simplicity, we will only consider $\mathsf{d} = \mathsf{q}$, thus $f_\eta = (f^* + \eta\mathsf{q})^* = \mathsf{M}_f^\eta$. The other assumption of S-APG in Nesterov (2005) is that f_η and the maximizer in its expression can be computed easily, which is precisely our Assumption 3.2. Finally for the general case where $\bar{f}$ is an average of K nonsmooth functions, the smoothing technique is applied in a component by component way, *i.e.*, approximate $\bar{f}$ with $\mathsf{M}_{\mathsf{A}^\eta}^\eta$.

It will be helpful to write down the key step in S-APG:

$$\mathbf{w}_{t+1} = \frac{\eta L_0}{1+\eta L_0}\left[\mathbf{w}_t - \frac{1}{L_0}\nabla\ell(\mathbf{w}_t)\right] + \frac{1}{1+\eta L_0}\sum_{k=1}^K \alpha_k \mathsf{P}_{f_k}^\eta(\mathbf{w}_t), \tag{3.14}$$

which is simply a convex combination of the usual gradient update over the smooth part ℓ and the proximal maps of the nonsmooth part $\{f_k\}$. For comparison, let us also repeat the key step in PA-APG:

$$\mathbf{w}_{t+1} = \sum_{k=1}^K \alpha_k \mathsf{P}_{f_k}^\eta\left(\mathbf{w}_t - \frac{1}{L_0}\nabla\ell(\mathbf{w}_t)\right). \tag{3.15}$$

Clearly, there is a striking similarity between the two algorithms. The "lag" in S-APG makes it more suitable for parallelization, in cases where *both* ℓ and $\bar{f}$ are sums of many components. On the other side, it is easy to see that S-APG finds a 2ϵ accurate solution in at most $O(\sqrt{L_0 + \overline{M^2}/(2\epsilon)}\sqrt{1/\epsilon})$ steps, since the Lipschitz constant of the gradient of $\ell + \mathsf{M}_{\mathsf{A}^\eta}^\eta$ is, under the choice of η in Theorem 3.1, upper bounded by $L_0 + \overline{M^2}/(2\epsilon)$. This is *strictly* worse than the $O(\sqrt{\max\{L_0, \overline{M^2}/(2\epsilon)\}}\sqrt{1/\epsilon})$ complexity of our approach. In other words, we have managed to remove the secondary term in the complexity bound of S-APG. We should emphasize that this strict improvement is obtained under exactly the same assumptions and with an algorithm as simple (if not simpler) as S-APG. In some sense it is quite remarkable that the seemingly "naive" approximation that pretends the linearity of the proximal map not only can be justified but also leads to a strictly better result.

Let us further explain how the improvement is possible. As mentioned, S-APG approximates $\bar{f}$ with the smooth function $\mathsf{M}_{\mathsf{A}^\eta}^\eta$. This smooth approximation is beneficial if our capability is limited to smooth functions. Put differently, S-APG implicitly treats applying the fast *gradient* algorithms as the ultimate goal. However, the recent advances on nonsmooth optimization have broadened the range of fast schemes: It is not smoothness but the proximal map that allows fast convergence. Just as how APG improves upon the subgradient method, our approach, with the ultimate goal to enable efficient computation of the proximal map, improves upon S-APG. Another lesson we wish to point

out is that unnecessary "over-smoothing", as in S-APG, does hurt the performance since it always increases the Lipschitz constant. To summarize, smoothing is not free and it should be used when truly needed.

Of course, the improved convergence rate is not entirely free. The current PA-APG cannot handle constraints (due to the Lipschitz assumption), therefore is not as general as S-APG. Secondly, we note that evaluating the function value of the proximal average might not be easy. This will create some issue when we need to perform a line search for the step size. An easy fix is to use approximate values as suggested by the inequality (3.11). On the flip side, the proximal average often approximates the original function strictly better than smoothing, see Figure 3.1 for an example.

Lastly, we note that our algorithm shares some similarity with forward-backward splitting procedures and alternating direction methods (Combettes and Pesquet 2011), although the exact connection would require nontrivial further work.

3.6 Some Refinements

This section contains several refinements of the basic idea in the previous sections.

3.6.1 Optimal weight

Firstly, using iv) in Proposition 3.1 we note that $\mathsf{M}^{\eta}_{\alpha f} = \alpha \mathsf{M}^{\alpha\eta}_{f} \neq \alpha \mathsf{M}^{\eta}_{f}$. In fact, for $\alpha \in]0,1[$, we have $\alpha \mathsf{M}^{\eta}_{f} \leq \mathsf{M}^{\eta}_{\alpha f} \leq \alpha f$. Therefore, it seems better to approximate the arithmetic average $\bar{f} := \sum_{i=1}^{K} \alpha_i f_i$ directly with $\sum_{i=1}^{K} \mathsf{M}^{\eta}_{\alpha_i f_i}$, rather than the Moreau average $\sum_{i=1}^{K} \alpha_i \mathsf{M}^{\eta}_{f_i}$. The same argument applies to the proximal average approximation. We now argue that as long as the weights $\boldsymbol{\alpha}$ are chosen in an optimal way, the two seemingly different approximations yield the same step size η, hence complexity bound, for APG or PG. Intuitively, this must be the case, as otherwise we could iterate the argument and keep improving the complexity bound, which is perhaps too good to be true.

Specifically, consider the *sum* regularizer $f = \sum_i f_i$, where as before each f_i is M_i-Lipschitz continuous *w.r.t.* the Hilbertian norm $\|\cdot\|_{\mathcal{H}}$. Let us apply the first approximation idea. Take the sum of Moreau envelops[7] $\sum_i \mathsf{M}^{1/\eta_i}_{f_i}$, which, by the duality between smoothness and strong convexity, has $(\sum_i \eta_i)$-Lipschitz continuous gradient. Therefore by Proposition 3.2, there exists some $h \in \Gamma_0$ such that $\mathsf{M}^{1/\sum_i \eta_i}_{h} = \sum_i \mathsf{M}^{1/\eta_i}_{f_i}$. We use h as our approximation to f. Similar as in Proposition 3.4, we have the uniform bound $0 \leq f - h \leq \sum_i M_i^2/(2\eta_i)$. As before, we tune down the step size $\eta = \min\{1/L_0, 1/\sum_i \eta_i\}$ so that there is no mismatch with the Lipschitz constant of the gradient of the smooth loss ℓ. Now, to get a 2ϵ-accurate solution, we need $\sum_i M_i^2/(2\eta_i) \leq \epsilon$, while to minimize the steps taken by APG or PG, we need η as large as possible, equivalently, $\sum_i \eta_i$ as small as possible. A simple application of the Cauchy-Schwarz inequality gives us the optimal choice $\eta_i = M_i(\sum_j M_j)/(2\epsilon)$, yielding $\eta = \min\{1/L_0, 2\epsilon/(\sum_i M_i)^2\}$.

[7] The reason to change the superscript from η to $1/\eta$ is to simplify the subsequent formula. Nothing magical.

Next consider the proximal average idea. Rewrite $f = \sum_i \alpha_i \cdot \tilde{f}_i$, where $\alpha_i > 0, \sum_i \alpha_i = 1$, and $\tilde{f}_i = f_i/\alpha_i$ is (M_i/α_i)-Lipschitz continuous. Take the Moreau average $\sum_i \alpha_i \mathsf{M}^\eta_{\tilde{f}_i}$, which equals to M^η_g for some $g \in \Gamma_0$ that is our proximal average approximation to f. Again, $0 \le f - g \le \sum_i \eta\alpha_i(M_i/\alpha_i)^2/2 = \sum_i \eta M_i^2/(2\alpha_i)$. To get a 2ϵ-accurate solution, we need $\eta \sum_i M_i^2/(2\alpha_i) \le \epsilon$ and the step size η as large as possible. Maximizing $\eta = \frac{\epsilon}{\sum_i M_i^2/(2\alpha_i)}$ *w.r.t.* $\alpha_i > 0, \sum_i \alpha_i = 1$ yields the same step size η, hence complexity bound, as in the previous paragraph, verifying our claim that the two approximations are essentially the same. In particular, we can set $\alpha_i = M_i/\sum_j M_j$ and $\eta = \frac{2\epsilon}{\sum_i M_i^2}$. As a by-product, we find that the optimal weight α_i simply balances out the Lipschitz constants of the component functions $\tilde{f}_i$, making perfect sense.

3.6.2 De-smoothing

Another fact that should become clear now is that the proximal average approximation amounts to de-smoothing the usual smooth approximation, that is, instead of using the smooth Moreau envelop $\sum_i \alpha_i \mathsf{M}^\eta_{f_i}$, we "pull" it back to a nonsmooth function A^η through the relation $\mathsf{M}^\eta_{\mathsf{A}^\eta} = \sum_i \alpha_i \mathsf{M}^\eta_{f_i}$. The benefit is obvious: we get a (strictly) tighter approximation without even increasing the Lipschitz constant of the gradient of the smooth part.

More generally, consider $f(\mathbf{z}) = \sum_i f_i(A_i\mathbf{z}) = \sum_i \alpha_i \tilde{f}_i(A_i\mathbf{z})$, where $f_i : \mathcal{H}' \to \mathbb{R} \cup \{\infty\}$ is convex and M_i-Lipschitz continuous, $A_i : \mathcal{H} \to \mathcal{H}'$ is some continuous linear operator, and $\tilde{f}_i = f_i/\alpha_i$. Due to the presence of the linear operator A_i, the proximal map $\mathsf{P}^\eta_{f_i \circ A_i}$ might not be easy to compute, even when $\mathsf{P}^\eta_{f_i}$ is given (unless A_i is say, orthonormal). In this case, we use the cruder approximation $\mathsf{M}^\eta_{\tilde{f}_i} \circ A_i$, whose derivative at $\mathbf{z}$ is easily seen to be $\frac{1}{\eta}A_i^\top[(\mathsf{Id} - \mathsf{P}^\eta_{\tilde{f}})(A_i\mathbf{z})] = \frac{1}{\eta}A_i^\top[(\mathsf{Id} - \mathsf{P}^1_{\eta\tilde{f}})(A_i\mathbf{z})] = \frac{1}{\eta}A_i^\top \mathsf{P}^1_{(\eta\tilde{f})^*}(A_i\mathbf{z})$. Clearly

$$\begin{aligned}
\left\| \tfrac{1}{\eta}A_i^\top \mathsf{P}^1_{(\eta\tilde{f})^*}(A_i\mathbf{z}_1) - \tfrac{1}{\eta}A_i^\top \mathsf{P}^1_{(\eta\tilde{f})^*}(A_i\mathbf{z}_2) \right\|_{\mathcal{H}} &\le \tfrac{1}{\eta}\left\|A_i^\top\right\| \cdot \left\| \mathsf{P}^1_{(\eta\tilde{f})^*}(A_i\mathbf{z}_1) - \mathsf{P}^1_{(\eta\tilde{f})^*}(A_i\mathbf{z}_2) \right\|_{\mathcal{H}'} \\
&\le \tfrac{1}{\eta}\left\|A_i^\top\right\| \cdot \|A_i\mathbf{z}_1 - A_i\mathbf{z}_2\|_{\mathcal{H}'} \\
&\le \tfrac{1}{\eta}\left\|A_i^\top\right\| \cdot \|A_i\| \cdot \|\mathbf{z}_1 - \mathbf{z}_2\|_{\mathcal{H}} \\
&= \tfrac{1}{\eta}\|A_i\|^2 \cdot \|\mathbf{z}_1 - \mathbf{z}_2\|_{\mathcal{H}},
\end{aligned}$$

with the induced norm on the linear operator A_i (or its adjoint $A_i^\top$). Therefore the approximation $\mathsf{M}^\eta_{\tilde{f}_i} \circ A_i$ has $(\frac{1}{\eta}\|A_i\|^2)$-Lipschitz continuous gradient. Taking the average and pulling back we know there exists $g \in \Gamma_0$ such that $\mathsf{M}^{\mu/\sum_i \alpha_i\|A_i\|^2}_g = \sum_i \alpha_i \mathsf{M}^\mu_{\tilde{f}_i} \circ A_i$. To get a 2ϵ-accurate solution, we need $\sum_i \alpha_i(M_i/\alpha_i)^2\mu/2 \le \epsilon$ and $\mu/\sum_i \alpha_i \|A_i\|^2$ as large as possible. Optimizing *w.r.t.* $\alpha_i > 0, \sum_i \alpha_i = 1$ and μ we obtain

$$\alpha_i = \frac{M_i/\|A_i\|}{\sum_j M_j/\|A_j\|}, \tag{3.16}$$

$$\mu = \frac{2\epsilon}{\sum_i M_i\|A_i\| \cdot \sum_j M_j/\|A_j\|}, \tag{3.17}$$

$$\eta := \min\{1/L_0, \mu/\sum_i \alpha_i \|A_i\|^2\} = \min\left\{1/L_0, \frac{2\epsilon}{(\sum_i M_i\|A_i\|)^2}\right\}. \tag{3.18}$$

Moreover, with the above parameters,

$$\mathsf{P}_g^\eta(\mathbf{z}) = \frac{\sum_i M_i \left\|A_i\right\| \left(\mathbf{z} - \tilde{A}_i^\top \tilde{A}_i \mathbf{z} + \tilde{A}_i^\top \mathsf{P}_{f_i}^{\eta_i}(\tilde{A}_i \mathbf{z})\right)}{\sum_j M_j \left\|A_j\right\|}, \tag{3.19}$$

where $\eta_i = \eta \left\|A_i\right\| \left(\sum_j \left\|A_j\right\| M_j\right)/M_i$ and $\tilde{A}_i = A_i / \left\|A_i\right\|$.

Finally, we can generalize the Moreau envelop to a non-Hilbertian setting by defining

$$\mathfrak{M}_f^\eta(\mathbf{z}) := \inf_{\mathbf{x}} f(\mathbf{x}) + \tfrac{1}{\eta}\mathsf{k}(\mathbf{z} - \mathbf{x}), \tag{3.20}$$

i.e. we convolve f with some kernel k that has Lipschitz continuous gradient *w.r.t.* some abstract norm $\|\cdot\|$. When f is Lipschitz continuous, we can again prove that $\mathfrak{M}_f^\eta$ is a uniform approximation to it. Whether or not we can de-smooth the smooth average $\sum_i \alpha_i \mathfrak{M}_{f_i}^\eta$ depends on the convexity of the range space $\bigcup_{f \in \Gamma_0} \{\mathfrak{M}_f^\eta\}$. Since a general theory necessarily involves a significant portion of technicality, we do not pursue the idea further here.

3.6.3 Nonsmooth Loss

We mentioned before that approximation itself does not lead to an inferior rate. Let us illustrate this point by applying the proximal average approximation to PSG which is discussed in Section 1.4.

Similar as before, we are interested in minimizing $\ell(\mathbf{w}) + \sum_{i=1}^K \alpha_i f_i(\mathbf{w})$ where f_i is M_i-Lipschitz continuous, $\alpha_i > 0, \sum_i \alpha_i = 1$. But this time we do not assume ℓ to be smooth; instead we require ℓ to be M_0-Lipschitz continuous. Simply run PSG on the uniform approximation

$$\ell(\mathbf{w}) + \mathsf{A}^\eta(\mathbf{w}), \tag{3.21}$$

with $\eta = \min\{c/M_0^2, 2/\overline{M^2}\}\epsilon$ for some $c \in]0, 2[$, then according to Corollary 1.2 we get an ϵ-accurate solution to (3.21) (as compared to any $\mathbf{w}$) after at most

$$\frac{M_0^2 \left\|\mathbf{w}_0 - \mathbf{w}\right\|_{\mathcal{H}}^2}{\min\{c, 2M_0^2/\overline{M^2}\}(2 - \min\{c, 2M_0^2/\overline{M^2}\})} \cdot \frac{1}{\epsilon^2}$$

steps, which is clearly on the same order as in Corollary 1.2. Moreover, due to our choice of the step size η, we actually have a 2ϵ-accurate solution to the original problem, *cf.* Proposition 3.4. The benefit, as compared to a vanilla implementation of PSG, is that we need only compute $\sum_i \alpha_i \mathsf{P}_{f_i}^\eta$ in each iteration, instead of the more troublesome $\mathsf{P}_{\sum_i \alpha_i f_i}^\eta$.

3.6.4 Varying Step Size

The current Theorem 3.1 assumes that some desired accuracy $\epsilon > 0$ is given *a priori*, and the (constant) step size η depends on it. We can remove this requirement by employing a varying step size η_t that deceases to 0 at a certain rate.

Indeed, we will analyze PA-APG as an example. First we need a technical result, which amounts to strengthening Proposition 3.3 and Proposition 3.4.

Proposition 3.5. *Let $f, g \in \Gamma_0$ and fix $\epsilon \geq 0$, we have*

i). $f \leq g + \epsilon \iff f^* \geq g^* - \epsilon \iff$ *for some, hence all* $\eta > 0$, $\mathsf{M}_f^\eta \leq \mathsf{M}_g^\eta + \epsilon$;

ii). Under Assumption 3.1, let $\eta \geq \lambda \geq 0$, *then* $\sum_i \alpha_i \mathsf{M}_{f_i}^\lambda \leq (\eta - \lambda)\frac{\overline{M^2}}{2} + \sum_i \alpha_i \mathsf{M}_{f_i}^\eta$;

iii). Both M_f^η *and* A^η *are decreasing* w.r.t. $\eta > 0$;

iv). Under Assumption 3.1, let $\eta \geq \lambda \geq 0$, *then* $\mathsf{A}^\lambda \leq (\eta - \lambda)\frac{\overline{M^2}}{2} + \mathsf{A}^\eta$.

Proof. i): The first equivalence is clear. Suppose now $\mathsf{M}_f^\eta \leq \mathsf{M}_g^\eta + \epsilon$ for some $\eta > 0$. Conjugating and use the first implication we have $f^* + \eta\mathsf{q} \geq g^* + \eta\mathsf{q} - \epsilon$, *i.e.*, $f^* \geq g^* - \epsilon$. Apply the first equivalence again we obtain $f \leq g + \epsilon$.

ii): It suffices to bound each component separately:

$$\mathsf{M}_{f_i}^\lambda - \mathsf{M}_{f_i}^\eta = (f_i^* + \lambda\mathsf{q})^* - \mathsf{M}_{f_i}^\eta = (f_i^* + \eta\mathsf{q} + (\lambda - \eta)\mathsf{q})^* - \mathsf{M}_{f_i}^\eta \leq (\eta - \lambda)M_i^2/2,$$

where the last inequality follows from i) and the observation that any point in $\operatorname{dom} f_i^*$ has norm at most M_i due to the Lipschitz assumption.

iii): From definition it is clear that M_f^η is decreasing *w.r.t.* η. The same claim about A^η can be seen using the rightmost formula in (3.9).

iv): Thanks to i), we need only prove $\mathsf{M}_{\mathsf{A}^\lambda}^\lambda \leq (\eta - \lambda)\frac{\overline{M^2}}{2} + \mathsf{M}_{\mathsf{A}^\eta}^\lambda$, which, due to iii), is further implied by $\mathsf{M}_{\mathsf{A}^\lambda}^\lambda \leq (\eta - \lambda)\frac{\overline{M^2}}{2} + \mathsf{M}_{\mathsf{A}^\eta}^\eta$. Now apply the definition of the proximal average and ii). □

Let us denote $F^{t+1} = \ell + \mathsf{A}^{\eta_t}$, then recalling some inequalities from the proof of Theorem 1.3, in particular,

$$\eta_t\gamma_{t+1}^2[F^{t+1}(\mathbf{w}_{t+1}) - F(\mathbf{w}^\star)] + \tfrac{1}{2}\left\|\mathbf{w}^\star - \mathbf{s}_{t+1}\right\|_{\mathcal{H}}^2 \leq \eta_t\gamma_t^2[F^{t+1}(\mathbf{w}_t) - F(\mathbf{w}^\star)] + \tfrac{1}{2}\left\|\mathbf{w}^\star - \mathbf{s}_t\right\|_{\mathcal{H}}^2,$$

with $\mathbf{w}^\star$ some (arbitrary) minimizer of $F = \ell + \bar{f}$. By the help of iv) in Proposition 3.5, we can further bound the right-hand side above:

$$\begin{aligned}
\eta_t\gamma_t^2[F^{t+1}(\mathbf{w}_t) - F(\mathbf{w}^\star)] &= \eta_t\gamma_t^2[F^{t+1}(\mathbf{w}_t) - F^t(\mathbf{w}_t)] + \eta_t\gamma_t^2[F^t(\mathbf{w}_t) - F(\mathbf{w}^\star)] \\
&\leq \eta_t\gamma_t^2(\eta_{t-1} - \eta_t)\overline{M^2}/2 + \eta_t\gamma_t^2[F^t(\mathbf{w}_t) - F(\mathbf{w}^\star)] \\
&= -\eta_t^2\gamma_t^2\overline{M^2}/2 + \eta_t\gamma_t^2[F^t(\mathbf{w}_t) - F(\mathbf{w}^\star) + \eta_{t-1}\overline{M^2}/2] \\
&\leq -\eta_t^2\gamma_{t+1}^2\overline{M^2}/2 + \eta_t^2\gamma_{t+1}\overline{M^2}/2 \\
&\quad + \eta_t\gamma_t^2[F^t(\mathbf{w}_t) - F(\mathbf{w}^\star) + \eta_{t-1}\overline{M^2}/2],
\end{aligned}$$

where the last inequality follows from the definition of γ_t. Now observe by iv) in Proposition 3.5 that $F^t(\mathbf{w}_t) - F(\mathbf{w}^\star) + \eta_{t-1}\overline{M^2}/2 \geq F(\mathbf{w}_t) - F(\mathbf{w}^\star) \geq 0$ since $\mathbf{w}^\star$ is optimal. Therefore by relaxing η_t to the bigger η_{t-1} we obtain a new recursion:

$$\eta_t\gamma_{t+1}^2[F^{t+1}(\mathbf{w}_{t+1}) - F(\mathbf{w}^\star) + \eta_t\overline{M^2}/2] + \tfrac{1}{2}\left\|\mathbf{w}^\star - \mathbf{s}_{t+1}\right\|_{\mathcal{H}}^2$$

$$\leq \eta_{t-1}\gamma_t^2[F^t(\mathbf{w}_t) - F(\mathbf{w}^\star) + \eta_{t-1}\overline{M^2}/2] + \tfrac{1}{2}\left\|\mathbf{w}^\star - \mathbf{s}_t\right\|_{\mathcal{H}}^2 + \eta_t^2\gamma_{t+1}\overline{M^2}/2.$$

Telescope and apply iv) in Proposition 3.5 once more:

$$F(\mathbf{w}_{T+1}) - F(\mathbf{w}^\star) \leq \frac{\overline{M^2}/2\sum_{t=0}^{T}\eta_t^2\gamma_{t+1} + \frac{1}{2}\left\|\mathbf{w}_0 - \mathbf{w}^\star\right\|_{\mathcal{H}}^2}{\eta_T\gamma_{T+1}^2}.$$

Verifying $\gamma_t = \Theta(t/2)$ and setting $\eta_t = \Theta(1/(t+1))$, we recover the $\tilde{O}(1/t)$ convergence rate as before, except a logarithmic factor hiding inside the big-O. It might be possible to further remove the logarithmic factor, at the cost of even more complications.

3.7 Experiments

We compare the proposed algorithm with S-APG on two important problems: overlapping group LASSO and graph-guided fused LASSO. See Example 1.7 and Example 1.9 for details about the nonsmooth function $\bar{f}$. We note that S-APG has been demonstrated with superior performance on both problems in Chen et al. (2012), therefore we will only concentrate on comparing with it.

Bear in mind that the purpose of our experiment is to verify the theoretical improvement as discussed in Section 3.5. We are not interested in fine tuning parameters here (despite its practical importance), thus for a fair comparison, we use the same desired accuracy ϵ, Lipschitz constant L_0 and other parameters for all methods. Since both our method and S-APG have the same per-step complexity, we will simply run them for a maximum number of iterations (after which saturation is observed) and report all the intermediate objective values.

Overlapping Group LASSO: Following Chen et al. (2012) we generate the data as follows: We set $\ell(\mathbf{w}) = \frac{1}{2\lambda K}\|A\mathbf{w} - \mathbf{b}\|^2$ where $A \in \mathbb{R}^{n\times d}$ whose entries are sampled from *i.i.d.* normal distributions, $w_j = (-1)^j \exp(-(j-1)/100)$, and $\mathbf{b} = A\mathbf{w} + \xi$ with the noise ξ sampled from the zero mean and unit variance normal distribution. Finally, the groups in the regularizer $\bar{f}$ are defined as

$$\{\{1,\ldots,100\},\{91,\ldots,190\},\ldots,\{d-99,\ldots,d\}\},$$

where $d = 90K + 10$. That is, there are K groups, each containing 100 variables, and the groups overlap by 10 consecutive variables. We adopt the uniform weight $\alpha_k = 1/K$, which is also optimal from the analysis in Section 3.6.1, and set the regularization parameter $\lambda = K/5$.

Figure 3.2 shows the results for $n = 5000$ and $K = 50$, with three different accuracy parameters. For completeness, we also include the results for the non-accelerated versions (PA-PG and S-PG). Clearly, accelerated algorithms are much faster than their non-accelerated cousins. Observe that our algorithms (PA-APG and PA-PG) converge consistently faster than S-APG and S-PG, respectively, with a big margin in the favorable case (middle panel). Again we emphasize that this improvement is achieved without any overhead.

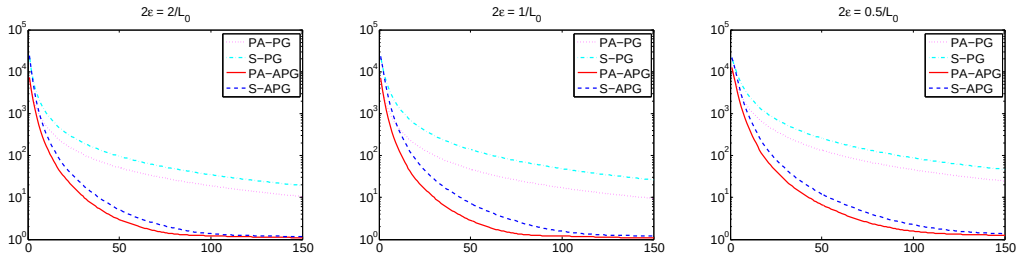


Figure 3.2: Objective value vs. iteration on overlapping group lasso.

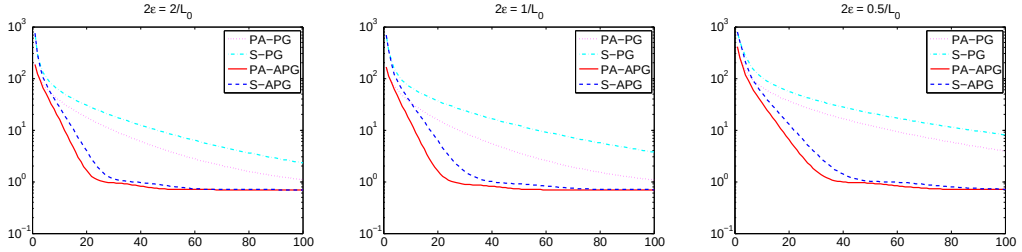


Figure 3.3: Objective value vs. iteration on graph-guided fused lasso.

Graph-guided Fused LASSO: We generate ℓ similarly as above. Following Chen et al. (2012), the graph edges E are obtained by thresholding the correlation matrix. The case $n = 5000, d = 1000, \lambda = 15$ is shown in Figure 3.3, under three different desired accuracies. Again, we observe that accelerated algorithms are faster than non-accelerated versions and our algorithms consistently converge faster.

3.8 Summary

We have considered the composite minimization problem which consists of a smooth loss and a sum of nonsmooth regularizers. This general framework encompasses many interesting machine learning applications. Unfortunately, the proximal map of the sum of regularizers is not easily computable, making fast schemes such as APG or PG hard to apply. However, based on the crucial observation that the proximal map of each individual regularizer is usually available in closed-form, we proposed a seemingly naive *nonsmooth* approximation which simply pretends the linearity of the proximal map. We justified our method using the proximal average, a new tool from convex analysis, and proved that the new approximation leads to a family of algorithms that strictly improves those based on the smoothing technique, which suffers from the increase of the Lipschitz constant. Several further refinements of the basic idea, including selecting an optimal weight, composing with a continuous linear map, handling nonsmooth loss, and varying step size, were presented. Lastly, experiments on both the overlapping group LASSO and the graph-guided fused LASSO confirmed the superiority of the proposed algorithms.

Chapter 4

Generalized Conditional Gradient

The main goal of this chapter is to develop yet another gradient algorithm. We are mostly motivated by the trace norm regularizer in matrix completion, whose proximal map, as we saw in Example 1.10, is available in closed-form but is nevertheless expensive to compute, making popular algorithms like PG or APG hard to apply in large-scale settings. Instead, the generalized conditional gradient (GCG) algorithm that we will thoroughly study in this chapter completely abandons the proximal map and turns to a linear subproblem which usually amounts to computing the polar of a norm regularizer. After a fairly complete overview of the existing GCG algorithm, with particular focus on its convergence properties, we propose a variant of it to handle positively homogeneous regularizers, since many useful regularizers in machine learning are of that form. We establish the $O(1/t)$ rate of convergence and discuss many theoretical properties of GCG. Then we present a simple relaxation strategy that turns the hard dictionary learning problem into a convex program, which our GCG variant can be easily deployed to optimize. To further accelerate the convergence, we intervene GCG with an effective (fixed-rank) local optimizer and we carefully show that the convergence property of GCG is still retained. Finally we verify the effectiveness of the proposed algorithm on two matrix learning problems.

The results in this chapter are mostly taken from Zhang et al. (2012), with some occasional mentioning of White et al. (2012); Zhang et al. (2011).

4.1 Generalized Conditional Gradient

Recall that our problem is to solve

$$\inf_{\mathbf{w}} F(\mathbf{w}), \quad \text{where} \quad F(\mathbf{w}) = \ell(\mathbf{w}) + f(\mathbf{w}). \tag{4.1}$$

We assume f is (closed, proper) convex and ℓ is continuously differentiable. We start with introducing the generalized conditional gradient algorithm in the general case where ℓ need not be convex, and then progressively we put in more assumptions and derive more interesting results.

We need a fair amount of facts from functional analysis, in particular many useful results about the "weak" topology in a Banach space. We do not repeat the related technical definitions but rec-

Algorithm 6 Generalized Conditional Gradient.

1: Initialize: $\mathbf{w}_0 \in \operatorname{dom} f$.
2: **for** $t = 0, 1, \ldots$ **do**
3: $\quad \mathbf{g}_t = \nabla \ell(\mathbf{w}_t)$
4: $\quad \mathbf{a}_t \in \operatorname{argmin}_{\mathbf{a}} \langle \mathbf{a}, \mathbf{g}_t \rangle + f(\mathbf{a})$
5: $\quad$ choose step size $\eta_t \in [0, 1]$
6: $\quad \tilde{\mathbf{w}}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t \mathbf{a}_t$
7: $\quad \mathbf{w}_{t+1} =$ `Update` ▷ Subroutine, see Definition 4.1
8: **end for**

ommend the very accessible[1] reference book of Trèves (1967). Or more conveniently (although less desirably), one can simply take the underlying space $\mathcal{H}$ to be a finite dimensional Euclidean space and interpret all "weak" topological notions as the familiar ones in a Euclidean space.

4.1.1 General Case

Like the proximal gradient algorithm, we motivate the development again from the perspective of operator splitting. Specifically, since ℓ is continuously differentiable and f is convex, $F = \ell + f$ is locally Lipschitz, therefore at a local extreme $\mathbf{w}$ we must satisfy the necessary condition

$$0 \in \partial F(\mathbf{w}) = \nabla \ell(\mathbf{w}) + \partial f(\mathbf{w}), \tag{4.2}$$

where ∂F denotes the generalized gradient[2] of Clarke (1990). Thus

$$-\nabla \ell(\mathbf{w}) \in \partial f(\mathbf{w}) \iff \mathbf{w} \in \partial f^*(-\nabla \ell(\mathbf{w})) \iff \mathbf{w} \in (1 - \eta)\mathbf{w} + \eta \partial f^*(-\nabla \ell(\mathbf{w})),$$

where f^* is the Fenchel conjugate of f, see Definition 1.3. So we have arrived at a fixed-point equation; hopefully repeated application of it will lead us at least to a stationary point. This is indeed so, as we will prove shortly.

The resulting procedure, called generalized conditional gradient (GCG), is summarized in Algorithm 6, where for clarity, we break the fixed-point iteration into several steps. The step size rule in line 5 is left unspecified until we formally state our convergence result. We have also inserted a subroutine `Update` in line 7; its purpose is to locally "improve" the iterate in the sense of the following definition:

Definition 4.1. *The subroutine* `Update` *is called* `Null` *if for all* t, $\mathbf{w}_{t+1} = \tilde{\mathbf{w}}_{t+1}$; `Descent` *if for all* t, $F(\mathbf{w}_{t+1}) \leq F(\tilde{\mathbf{w}}_{t+1})$; `Relaxed` *if for all* t,

$$F(\mathbf{w}_{t+1}) \leq \ell(\mathbf{w}_t) + \eta_t \langle \mathbf{a}_t - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \rangle + \tfrac{L\eta_t^2}{2} \|\mathbf{a}_t - \mathbf{w}_t\|^2 + (1 - \eta_t) f(\mathbf{w}_t) + \eta_t f(\mathbf{a}_t),$$

where L *is some constant that will be introduced later.*

[1] By this we mean both intellectually and economically.
[2] It is *not* an abuse of notation to denote both the generalized gradient and the subdifferential by ∂ since it can be proved that the two are the same when convexity is present.

Note that we allow the subroutine `Update` *to access both* ℓ *and* f*, if needed, therefore it can be very powerful. We deliberately do not specify the inputs to* `Update`*, so as to signal this flexibility, which, if kept in mind, will help us shorten and unify some proofs.*

For example, using a `Descent` subroutine to force $F(\mathbf{w}_{t+1}) = \min\{F(\mathbf{w}_t), F(\tilde{\mathbf{w}}_{t+1})\}$ will ensure a monotonic decrease of the objective. The `Relaxed` subroutine is introduced mainly as a proof means; as we will see, any `Descent` subroutine is `Relaxed` if $\nabla\ell$ satisfies a Lipschitz condition. For later reference we also record the most important step, line 4, here:

$$\mathbf{a}_t \in \{\operatorname*{argmin}_{\mathbf{a}} \langle \mathbf{a}, \nabla\ell(\mathbf{w}_t)\rangle + f(\mathbf{a})\} = \partial f^*(-\nabla\ell(\mathbf{w}_t)). \tag{4.3}$$

In words, in each iteration we linearize the smooth loss ℓ, solve the subproblem (4.3), select the step size, take the convex combination, and finally commit a local improvement. The subproblem (4.3) shares some similarity with the proximal map that we studied in Chapter 2: both choose to leave the potentially nonsmooth function f untouched. The difference is also apparent: we replace the smooth loss ℓ with a linear term rather than a quadratic term. As a consequence, the subproblem (4.3) may have multiple solutions in which case we simply contend with any one of them, or no solution for which we will pose extra assumptions to avoid. Another major difference is that GCG, by definition, can be "run" in any topological vector space while other algorithms, such as PG or APG, are more "picky" about the underlying space (at least the topology is "strong" enough to hold strongly convex functions).

To the best of our knowledge, GCG is first studied by Mine and Fukushima (1981) in a finite dimensional setting and later by Bredies et al. (2009) in the Hilbertian setting. The latter also suggested the name GCG[3]. GCG naturally generalizes the old conditional gradient which was first studied by Frank and Wolfe (1956) in the case $f = \iota_C$ for some polyhedral set C and then by Dem'yanov and Rubinov (1967); Levitin and Polyak (1966) in the case $f = \iota_C$ for any closed and bounded set C.

Let us be precise about the assumptions we need.

Assumption 4.1. *ℓ is continuously differentiable in an open set that contains* $\operatorname{dom} f$*; f is (closed, proper) convex with*[4] $-\nabla\ell(\operatorname{dom} f) \subseteq \operatorname{Range}(\partial f)$.

The range assumption simply makes sure that the subproblem (4.3) has at least one solution.

A useful quantity that we will need in the proof is the duality gap:

$$\mathsf{G}(\mathbf{w}) := \langle \mathbf{w}, \nabla\ell(\mathbf{w})\rangle + f(\mathbf{w}) - \min_{\mathbf{a}} \langle \mathbf{a}, \nabla\ell(\mathbf{w})\rangle + f(\mathbf{a}). \tag{4.4}$$

By definition $\mathsf{G}(\mathbf{w}) \geq 0$ and equality holds if $\mathbf{w}$ satisfies the necessary condition (4.2). Moreover, if ℓ is convex,

$$\mathsf{G}(\mathbf{w}) = \max_{\mathbf{a}} \langle \mathbf{w} - \mathbf{a}, \nabla\ell(\mathbf{w})\rangle + f(\mathbf{w}) - f(\mathbf{a}) \geq \max_{\mathbf{a}} F(\mathbf{w}) - F(\mathbf{a}),$$

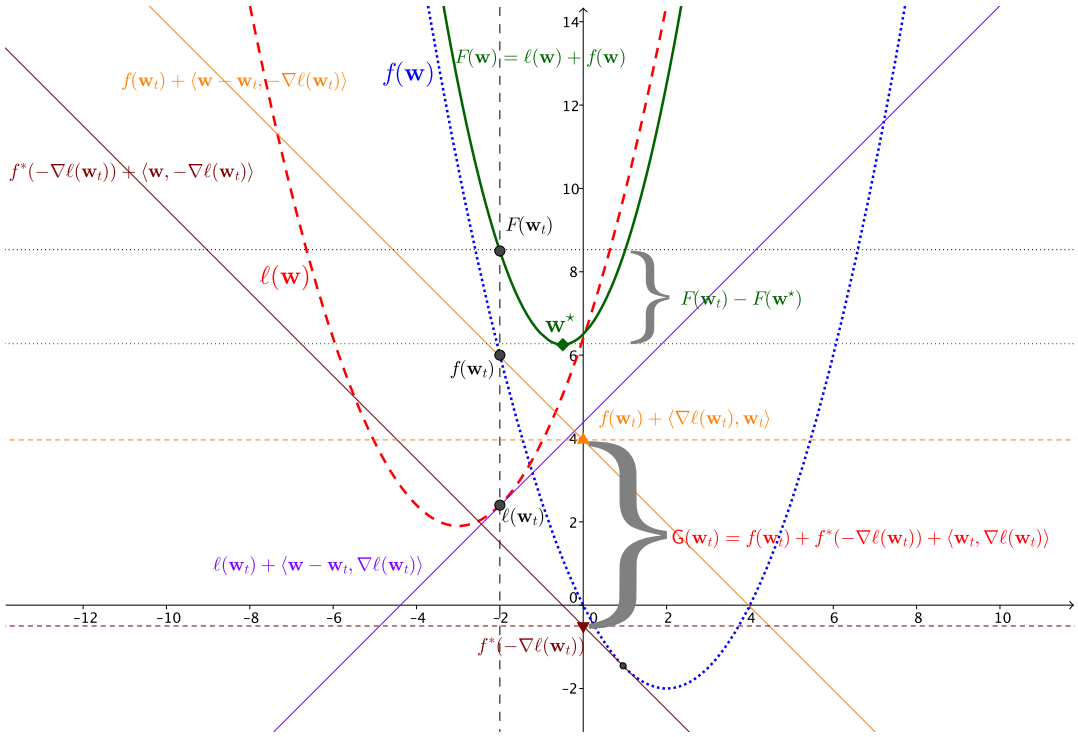


Figure 4.1: Here both ℓ (the red dashed parabolic) and f (the blue dotted parabolic) are convex (quadratic) functions, and $\mathbf{w}^\star$ denotes the minimizer of $F = \ell + f$ (the green solid parabolic). The dashed vertical line represents our current iterate $\mathbf{w}_t$. As predicted, the duality gap $\mathsf{G}(\mathbf{w}_t) \geq F(\mathbf{w}_t) - F(\mathbf{w}^\star)$.

upper bounding the suboptimality of $F(\mathbf{w})$, see Figure 4.1. Note that $\mathsf{G}(\mathbf{w}_t)$ can be computed as a by-product in each iteration of Algorithm 6, therefore it can be used as a natural stopping criteria.

We need two more assumptions.

Assumption 4.2. *The underlying space $\mathcal{H}$ is Banach and $\nabla\ell$ is uniformly continuous on bounded sets.*

Clearly, when $\mathcal{H}$ is of finite dimension, Assumption 4.2 is automatically satisfied under Assumption 4.1. The uniform continuity on $\nabla\ell$ is also self-granted if $\nabla\ell$ is Lipschitz continuous (on bounded sets).

Assumption 4.3. *$\{\mathbf{a}_t\}_t$ and $\{\mathbf{w}_t\}_t$ generated by the algorithm are bounded.*

This assumption is more stringent and we will discuss it after our first convergence result about Algorithm 6:

Theorem 4.1. *Under Assumption 4.1, Algorithm 6, equipped with a* `Descent` *subroutine and the*

[3]This name seems particularly fitting since the algorithm naturally *generalizes* the old conditional gradient based on Clarke's *generalized* gradient.

[4]Naturally by $-\nabla\ell(\operatorname{dom} f)$ we mean the set $\bigcup_{\mathbf{w}\in\operatorname{dom} f}\{-\nabla\ell(\mathbf{w})\}$.

step size η_t satisfying

$$F(\tilde{\mathbf{w}}_{t+1}) \le \min_{0\le\eta\le 1} \ell((1-\eta)\mathbf{w}_t + \eta\mathbf{a}_t) + (1-\eta)f(\mathbf{w}_t) + \eta f(\mathbf{a}_t), \tag{4.5}$$

yields, in each time step t, either $F(\mathbf{w}_{t+1}) < F(\mathbf{w}_t)$ or $\mathsf{G}(\mathbf{w}_t) = 0$.

Additionally, if Assumption 4.2 and Assumption 4.3 hold, then either $F(\mathbf{w}_t) \downarrow -\infty$, or $\mathsf{G}(\mathbf{w}_t) = 0$ indefinitely, or $\mathsf{G}(\mathbf{w}_t) \to 0$.

Proof. Due to the step size rule, the algorithm always makes monotonic progress. We now strengthen this observation by looking more carefully at the step size rule. Since the subroutine is `Descent`,

$$\begin{aligned} F(\mathbf{w}_{t+1}) \le F(\tilde{\mathbf{w}}_{t+1}) &:= F((1-\eta_t)\mathbf{w}_t + \eta_t\mathbf{a}_t) \\ &\le \min_{0\le\eta\le 1} \ell((1-\eta)\mathbf{w}_t + \eta\mathbf{a}_t) + (1-\eta)f(\mathbf{w}_t) + \eta f(\mathbf{a}_t) \\ &= \min_{0\le\eta\le 1} \ell(\mathbf{w}_t) + \eta\left\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{u}_t)\right\rangle + (1-\eta)f(\mathbf{w}_t) + \eta f(\mathbf{a}_t) \\ &= \min_{0\le\eta\le 1} F(\mathbf{w}_t) + \eta[\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{u}_t)\rangle - f(\mathbf{w}_t) + f(\mathbf{a}_t)], \end{aligned} \tag{4.6}$$

where, using the mean value theorem, $\mathbf{u}_t$ is some vector lying between $\mathbf{w}_t$ and $(1-\eta)\mathbf{w}_t + \eta\mathbf{a}_t$. As $\eta \to 0$, $\mathbf{u}_t \to \mathbf{w}_t$, hence by continuity, $\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{u}_t)\rangle - f(\mathbf{w}_t) + f(\mathbf{a}_t) \to -\mathsf{G}(\mathbf{w}_t) \le 0$. If $\mathsf{G}(\mathbf{w}_t) = 0$ we have nothing to prove, otherwise we have $F(\mathbf{w}_{t+1}) < F(\mathbf{w}_t)$.

In the rest of the proof we assume that $\mathsf{G}(\mathbf{w}_t) \neq 0$ for all t sufficiently large and that $F(\mathbf{w}_t)$ converges to a finite limit. Rearrange (4.6):

$$F(\mathbf{w}_{t+1}) - F(\mathbf{w}_t) \le \min_{0\le\eta\le 1} \eta[\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{u}_t) - \nabla\ell(\mathbf{w}_t)\rangle - \mathsf{G}(\mathbf{w}_t)].$$

By assumption $\{\mathbf{a}_t\}_t, \{\mathbf{w}_t\}_t$ are bounded, thus $\{\mathbf{u}_t\}_t$ is bounded too (as $\mathbf{u}_t$ is some convex combination of $\mathbf{a}_t$ and $\mathbf{w}_t$). On the other hand, since $\nabla\ell$ is assumed to be uniformly continuous on bounded sets, when η is sufficiently small, say $\eta \le \tilde{\eta} > 0$, we have $\mathbf{u}_t$ sufficiently close to $\mathbf{w}_t$ such that

$$\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{u}_t) - \nabla\ell(\mathbf{w}_t)\rangle \le \|\nabla\ell(\mathbf{u}_t) - \nabla\ell(\mathbf{w}_t)\|_\circ \cdot \|\mathbf{a}_t - \mathbf{w}_t\| \le \epsilon,$$

for some (arbitrary) $\epsilon > 0$. Crucially, $\tilde{\eta}$ does not depend on t, thanks to the boundedness and the uniform continuity assumption. Therefore for t sufficiently large we have

$$\tilde{\eta}[\mathsf{G}(\mathbf{w}_t) - \epsilon] \le F(\mathbf{w}_t) - F(\mathbf{w}_{t+1}) \le \epsilon\tilde{\eta},$$

that is, $\mathsf{G}(\mathbf{w}_t) \le 2\epsilon$. Since ϵ can be made as small as we please, the proof is complete. □

The boundedness of $\{\mathbf{a}_t\}_t$ and $\{\mathbf{w}_t\}_t$—the key in the proof—can be forced under a set of different conditions, and we summarize some popular ones as follows.

Proposition 4.1. *Under Assumption 4.1 and Assumption 4.2, let the subroutine be `Descent`, then Assumption 4.3 is satisfied if either of the following holds:*

a). The sublevel set $\{\mathbf{w} \in \operatorname{dom} f : F(\mathbf{w}) \le F(\mathbf{w}_0)\}$ is compact, and $-\nabla\ell(\operatorname{dom} f) \subseteq \operatorname{int}(\operatorname{dom} f^*)$.

The latter condition holds, in particular, when f is cofinite, i.e.*, f^* has full domain;*

b). The sublevel set $\{\mathbf{w} \in \operatorname{dom} f : F(\mathbf{w}) \leq F(\mathbf{w}_0)\}$ *is bounded, and* f *is super-coercive,* i.e., $\lim_{\|\mathbf{w}\|\to\infty} f(\mathbf{w})/\|\mathbf{w}\| \to \infty$*;*

c). $\operatorname{dom} f$ *is bounded.*

Proof. a): Let C be the closure of the sequence $\{\mathbf{w}_t\}_t$. Due to the compactness assumption on the sublevel set and the monotonicity of $F(\mathbf{w}_t)$, C is compact. Moreover $C \subseteq \operatorname{dom} f$ since for all cluster point, say $\mathbf{w}$, of $\mathbf{w}_t$ we have from the closedness of F that $F(\mathbf{w}) \leq \liminf F(\mathbf{w}_{t_k}) \leq F(\mathbf{w}_0) < \infty$. Since $-\nabla\ell$ is continuous, $-\nabla\ell(C)$ is a compact subset of $\operatorname{int}(\operatorname{dom} f^*)$. Note that f^* is continuous on the interior of its domain[5], therefore its subdifferential is locally bounded on $-\nabla\ell(C)$, see *e.g.* Borwein and Vanderwerff (2010, Proposition 4.1.26). A standard compactness argument then establishes the boundedness of $(\partial f^*)(-\nabla\ell(C))$. Thus $\{\mathbf{a}_t\}_t$ is bounded.

b): Note first that the boundedness of $\{\mathbf{w}_t\}_t$ follows immediately from the boundedness assumption on the sublevel set, thanks to the monotonic property of $F(\mathbf{w}_t)$. Since $\nabla\ell$ is uniformly continuous, the set $\{-\nabla\ell(\mathbf{w}_t)\}_t$ is again bounded. On the other hand, we know from Borwein and Vanderwerff (2010, Theorem 4.4.13, Proposition 4.1.25) that f is super-coercive iff ∂f^* maps bounded sets into bounded sets. Therefore $\{\mathbf{a}_t\}_t$ is again bounded.

c): Clearly meets b). □

The three conditions above (in slightly restricted forms) appeared in (Mine and Fukushima 1981), (Bredies et al. 2009), and (Dem'yanov and Rubinov 1967; Frank and Wolfe 1956; Levitin and Polyak 1966), respectively. Note that under condition a) we actually know that F is bounded from below while under condition b) if $\mathcal{H}$ is reflexive (such as Hilbertian) and F is convex (or weakly closed), then again F is bounded from below. It is interesting to compare condition a) and b): There appears to be a trade-off between the assumption on the sublevel set of F and the assumption on the behavior of f^*. In particular, super-coercive implies cofinite while the converse is only true in finite dimensions (Borwein and Vanderwerff 2010).

Some further remarks about Theorem 4.1 are in order.

Remark 4.1. *Due to the possible non-uniqueness in the subproblem (4.3),* $\mathsf{G}(\mathbf{w}_t) = 0$ *does not imply* $\mathsf{G}(\mathbf{w}_s) = 0$ *for all* $s > t$*. However, when* F *is convex,* $\mathsf{G}(\mathbf{w}_t) = 0$ *implies* $\mathbf{w}_t$ *is globally optimal, in which case we do have* $\mathsf{G}(\mathbf{w}_s) = 0$ *for all* $s > t$ *since the monotonicity* $F(\mathbf{w}_s) \leq F(\mathbf{w}_t)$ *implies the global optimality of* $\mathbf{w}_s$*. On the other hand, if* f *is strictly convex, then hitting* $\mathsf{G}(\mathbf{w}_t) = 0$ *for some* t *implies* $\mathbf{a}_t = \mathbf{w}_t$ *hence* $\mathbf{w}_s = \mathbf{w}_t$ *for all* $s \geq t$*, provided that we employ the* `Null` *subroutine in Algorithm 6.*

Remark 4.2. *It is easily seen that the duality gap* G *is lower semicontinuous, therefore if any subsequence of* $\mathbf{w}_t$ *converges to some point, say* $\mathbf{w}$*, then we have* $0 \leq \mathsf{G}(\mathbf{w}) \leq \liminf \mathsf{G}(\mathbf{w}_{t_k}) = 0$, i.e.*, we indeed converge to a stationary point. Of course, when* F *has compact level sets (such as a)*

[5] This is where we *do* need the completeness of the underlying space.

in Proposition 4.1), $\mathbf{w}_t$ is guaranteed to have a convergent subsequence. Weak convergence can be argued similarly (under slightly different assumptions).

Remark 4.3. *Of the few small improvements we made in Theorem 4.1, as compared with Bredies et al. (2009); Mine and Fukushima (1981), we would like to emphasize the step size rule (4.5). Previous work insisted on picking*

$$\eta_t \in \operatorname*{argmin}_{0\leq\eta\leq 1} \ell((1-\eta)\mathbf{w}_t + \eta\mathbf{a}_t) + f((1-\eta)\mathbf{w}_t + \eta\mathbf{a}_t), \tag{4.7}$$

which clearly is a special case of our rule (4.5), thanks to the convexity of f. Our observation of the sufficiency of (4.5), although quite straightforward, may have a major algorithmic consequence: minimizing the right-hand side of (4.5) can be significantly easier than dealing with (4.7) directly. Indeed, in their application to a linear inverse problem, Bredies et al. (2009) had to develop specialized subroutines (under further assumptions on some parameter p) for solving (4.7), while the right-hand side of (4.5) would be trivial to apply in their setting (without any assumption on the parameter p). Of course, from the point of view of greedily decreasing the objective value, (4.7) is the best among all possibilities of the general rule (4.5).

Perhaps more surprisingly, a careful inspection of the proof reveals that we have not *used the convexity assumption on f* explicitly *anywhere! Convexity is implicitly needed only in two places: the tractability of the subproblem (4.3) and the satisfiability of the step size rule (4.5).*

4.1.2 Lipschitz Case

In this section, we push Theorem 4.1 harder under a slightly more restrictive assumption:

Assumption 4.4. *There exists some positive constant $L < \infty$ such that for the sequence $\{\mathbf{w}_t\}_t, \{\mathbf{a}_t\}_t$ generated by the algorithm and for all $\eta \in [0,1]$, we have*

$$\ell(\mathbf{w}_t + \eta(\mathbf{a}_t - \mathbf{w}_t)) \leq \ell(\mathbf{w}_t) + \eta \langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{w}_t)\rangle + \tfrac{L\eta^2}{2} \|\mathbf{a}_t - \mathbf{w}_t\|^2 . \tag{4.8}$$

The inequality (4.8) is exactly the one in Assumption 1.2 of Chapter 1, under slight disguise. In fact, all we need is the weaker inequality (which itself does not even require a norm, or topology)

$$\ell(\mathbf{w}_t + \eta(\mathbf{a}_t - \mathbf{w}_t)) \leq \ell(\mathbf{w}_t) + \eta \langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{w}_t)\rangle + \tfrac{L_F\eta^2}{2}, \tag{4.9}$$

for some positive constant $L_F < \infty$. Indeed, let ρ be the smallest number so that the ball with radius ρ contains the sequence $\{\mathbf{w}_t\}_t, \{\mathbf{a}_t\}_t$, then $L_F \leq L\rho^2$, provided that ℓ satisfies Assumption 4.4 with constant L. As mentioned in Chapter 1, (4.8) holds as long as the gradient $\nabla\ell$ is L-Lipschitz continuous *w.r.t.* the dual norm $\|\cdot\|_\circ$, in which case Assumption 4.2 also becomes trivial. The reason to use the stronger condition (4.8) is that we do not need to upper bound $\|\mathbf{a}_t - \mathbf{w}_t\|$ *a priori*, which, although can be done under Proposition 4.1, is usually loose. A second reason is that almost all examples we are aware of deduce (4.9) from (4.8). Clearly under Assumption 4.1 and Assumption 4.4, a `Descent` subroutine is automatically `Relaxed`.

We are now ready to state the sharpened

Theorem 4.2. *Under Assumption 4.1, Assumption 4.2, Assumption 4.3 and Assumption 4.4, Algorithm 6 with the subroutine* `Null` *and the step size rule*

$$\eta_t = \min\left\{\frac{\mathsf{G}(\mathbf{w}_t)}{L\left\|\mathbf{w}_t - \mathbf{a}_t\right\|^2}, 1\right\}, \tag{4.10}$$

yields either $F(\mathbf{w}_t) \downarrow -\infty$ *or* $G(\mathbf{w}_t) \to 0$.

Proof. Not surprisingly the step size is chosen to minimize the quadratic upper bound:

$$\begin{aligned} F(\mathbf{w}_{t+1}) &\leq \min_{\eta\in[0,1]} \ell(\mathbf{w}_t) + \eta\left\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{w}_t)\right\rangle + \tfrac{L\eta^2}{2}\left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2 + (1-\eta)f(\mathbf{w}_t) + \eta f(\mathbf{a}_t) \\ &= \min_{\eta\in[0,1]} F(\mathbf{w}_t) - \eta\mathsf{G}(\mathbf{w}_t) + \tfrac{L\eta^2}{2}\left\|\mathbf{w}_t - \mathbf{a}_t\right\|^2. \end{aligned}$$

Therefore if $G(\mathbf{w}_t) > 0$, through minimizing η in the above we have $F(\mathbf{w}_{t+1}) < F(\mathbf{w}_t)$. On the other hand, if $\mathsf{G}(\mathbf{w}_t) = 0$ for some t, then $\eta_t = 0$, resulting in $\mathbf{w}_{t+1} = \mathbf{w}_t$. Thus the algorithm will not change its iterate afterwards.

Assume that $\mathsf{G}(\mathbf{w}_t) \neq 0$ for any t and that $F(\mathbf{w}_t)$ converges to a finite limit (otherwise there is nothing to prove). Analyzing the step size in each case separately, we have

$$\min\left\{\frac{\mathsf{G}^2(\mathbf{w}_t)}{2L\left\|\mathbf{w}_t - \mathbf{a}_t\right\|^2}, \frac{\mathsf{G}(\mathbf{w}_t)}{2}\right\} \leq F(\mathbf{w}_t) - F(\mathbf{w}_{t+1}) \to 0.$$

Due to the boundedness assumption in Assumption 4.3, we know $\mathsf{G}(\mathbf{w}_t) \to 0$. □

Theorem 4.2, with condition c) of Proposition 4.1 to ensure Assumption 4.3, appeared in Levitin and Polyak (1966, Theorem 6.1 (1)). The extension to a general f (that is not necessarily an indicator function) does not pose any difficulty.

Next, let us look at a non-adaptive choice of the step size rule:

Theorem 4.3. *Under Assumption 4.1, Assumption 4.2, Assumption 4.3 and Assumption 4.4, Algorithm 6 with the subroutine* `Relaxed` *and the subproblem* (4.3) *being solved up to some additive error* ε_t *yields*

$$\sum_{s=0}^{t}\left[\eta_s\mathsf{G}(\mathbf{w}_s) - \tfrac{L\eta_s^2}{2}\left\|\mathbf{w}_s - \mathbf{a}_s\right\|^2 - \eta_s\varepsilon_s\right] \leq F(\mathbf{w}_0) - F(\mathbf{w}_{t+1}). \tag{4.11}$$

Moreover, if F *is bounded from below,* $\sum_t \eta_t = \infty, \sum_t \eta_t^2 < \infty$, *and* $\varepsilon_t = O(1/H_t^{1+\delta})$ *for some* $\delta > 0$, *then* $\liminf_{t\to\infty} \mathsf{G}(\mathbf{w}_t)H_t = 0$, *where* $H_t = \sum_{s=0}^{t}\eta_s$ *is the partial sum.*

Proof. Since the subroutine is `Relaxed`,

$$\begin{aligned} F(\mathbf{w}_{t+1}) &\leq \ell(\mathbf{w}_t) + \eta_t\left\langle \mathbf{a}_t - \mathbf{w}_t, \nabla\ell(\mathbf{w}_t)\right\rangle + \tfrac{L\eta_t^2}{2}\left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2 + (1-\eta_t)f(\mathbf{w}_t) + \eta_t f(\mathbf{a}_t) \\ &= F(\mathbf{w}_t) - \eta_t\mathsf{G}(\mathbf{w}_t) + \eta_t\varepsilon_t + \tfrac{L\eta_t^2}{2}\left\|\mathbf{w}_t - \mathbf{a}_t\right\|^2. \end{aligned}$$

Rearranging and telescoping leads to (4.11).

The second claim follows simply from the observation that $\sum_t \eta_t \mathsf{G}(\mathbf{w}_t)$ is bounded from above under the given assumptions. Specifically, note that the divergence of the partial sum $H_t := \sum_{s=0}^{t} \eta_s \to \infty$ implies that $\sum_t \eta_t / H_t \to \infty$ and $\sum_t \eta_t / H_t^{1+\delta} < \infty$ for any $\delta > 0$, see *e.g.* Hardy et al. (1952, Result 162). □

Apparently there is a trade-off between the asymptotic rate of $\mathsf{G}(\mathbf{w}_t)$ approaching 0 and the error tolerance in each subproblem (4.3). Of course there are many admissible choices of the step size, for instance $\eta_t = O(1/t^\beta)$ for any $1/2 < \beta \leq 1$ would do. A slight advantage of the non-adaptive step size rule, as compared to the "optimal" one (4.10) obtained from minimizing the quadratic upper bound, is that it does not need the constant L explicitly; a warrant of the existence of L suffices. The error tolerant property is *not* specific to the non-adaptive step size; it is possible to have it in Theorem 4.1 and Theorem 4.2 too.

Our final result in this section is about the convergence of the iterates $\{\mathbf{w}_t\}_t$. For this we need a different assumption.

Assumption 4.5. *The underlying space $\mathcal{H}$ is Hilbertian, $\nabla \ell$ is L-Lipschitz continuous, and f is L-strongly convex,* i.e., *$f - \frac{L}{2} \|\cdot\|_{\mathcal{H}}^2$ is convex.*

As mentioned before, this assumption automatically implies both Assumption 4.2 and Assumption 4.4, provided that Assumption 4.3 holds, which does as we will see.

Theorem 4.4. *Under Assumption 4.1, Assumption 4.5, and assuming that* $\operatorname{dom} f$ *is closed, that the subroutine is* `Null`*, that F has at least one stationary point (*i.e. *some* $\mathbf{w}$ *such that* $\mathbf{0} \in \partial F(\mathbf{w})$*), and that the step size $\eta_t \in [0, 1]$ satisfies $\sum_t \eta_t(1 - \eta_t) = \infty$, then the iterates $\{\mathbf{w}_t\}_t$ generated by Algorithm 6 converge weakly to some stationary point $\mathbf{w}^\star$.*

Proof. Indeed, define $D = \operatorname{dom} f$ which is closed and convex by assumption, and define $T(\mathbf{w}) := \partial f^*(\nabla \ell(\mathbf{w}))$. As discussed in Chapter 3, the assumption of f being L-strongly convex actually implies that f^* is differentiable with $1/L$-Lipschitz continuous gradient. Thus $T : D \to D$, being the composition of an L-Lipschitz continuous function $\nabla \ell$ and a $1/L$-Lipschitz continuous function ∇f^*, is nonexpansive. In our motivation of GCG (at the beginning of Section 4.1.1), we pointed out that it is nothing but the fixed-point iteration $\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t T(\mathbf{w}_t)$. Therefore the claim follows immediately from the well-known Krasnosel'skiĭ-Mann theorem, see *e.g.* Bauschke and Combettes (2011, Theorem 5.14). □

Surprisingly, Theorem 4.4, in the current context, appears to be new, despite its directness. We remark that as a consequence of the above theorem, Assumption 4.3 is also automatically met. Indeed, $\{\mathbf{w}_t\}_t$, as a weakly convergent sequence, is bounded, and the strong convexity of f implies super-coercive, therefore a similar argument as in condition b) of Proposition 4.1 establishes Assumption 4.3. Clearly, any step size rule such that $\eta_t = \Omega(1/t)$ works for us.

4.1.3 Convex Case

So far we have satisfied ourselves with the convergence to a stationary point, due to the apparent generality we have enjoyed. In this section we delve into the convex setting so that shaper results on the convergence rate can be derived.

Theorem 4.5. *Under Assumption 4.1 and Assumption 4.4, and assume that ℓ is convex, that the subroutine is* `Relaxed`*, and that the subproblem (4.3) is solved up to some additive error $\varepsilon_t \geq 0$, then we have for any $\mathbf{w} \in \operatorname{dom} F$, Algorithm 6 yields*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}) + \pi_t(1-\eta_0)(F(\mathbf{w}_0) - F(\mathbf{w})) + \sum_{s=0}^{t} \frac{\pi_t}{\pi_s} \frac{\eta_s^2}{2}(2\varepsilon_s/\eta_s + L\left\|\mathbf{a}_s - \mathbf{w}_s\right\|^2), \quad (4.12)$$

where $\pi_t := \prod_{s=1}^{t}(1-\eta_s)$ with $\pi_0 = 1$.

Moreover, the minimal duality gap $\tilde{\mathsf{G}}_t := \min_{k+1 \leq s \leq t} \mathsf{G}(\mathbf{w}_s)$ satisfies, for all $k \geq 0$,

$$\tilde{\mathsf{G}}_t \leq \frac{1}{\sum_{s=k+1}^{t} \eta_s} \left[F(\mathbf{w}_{k+1}) - F(\mathbf{w}) + \sum_{s=k+1}^{t} \frac{\eta_s^2}{2}(2\varepsilon_s/\eta_s + L\left\|\mathbf{a}_s - \mathbf{w}_s\right\|^2) \right]. \quad (4.13)$$

Proof. Since the subroutine is `Relaxed`,

$$\begin{aligned} F(\mathbf{w}_{t+1}) &\leq \ell(\mathbf{w}_t) + \eta_t \left\langle \mathbf{a}_t - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \right\rangle + \tfrac{L\eta_t^2}{2} \left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2 \\ &\quad + (1-\eta_t) f(\mathbf{w}_t) + \eta_t f(\mathbf{a}_t) \\ \%\ \varepsilon_t\text{-optimality of } \mathbf{a}_t\ \% &\leq F(\mathbf{w}_t) - \eta_t \mathsf{G}(\mathbf{w}_t) + \tfrac{L\eta_t^2}{2} \left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2 + \eta_t \varepsilon_t \\ &= F(\mathbf{w}_t) - \eta_t \mathsf{G}(\mathbf{w}_t) + \tfrac{\eta_t^2}{2}(2\varepsilon_t/\eta_t + L\left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2). \end{aligned}$$

Define $\Delta_t := F(\mathbf{w}_t) - F(\mathbf{w})$ and $\mathsf{G}_t := \mathsf{G}(\mathbf{w}_t)$. Thus

$$\Delta_{t+1} \leq \Delta_t - \eta_t \mathsf{G}_t + \tfrac{\eta_t^2}{2}(2\varepsilon_t/\eta_t + L\left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2), \quad (4.14)$$

$$\Delta_t \leq \mathsf{G}_t. \quad (4.15)$$

Plug (4.15) into (4.14) and expand:

$$\Delta_{t+1} \leq \pi_t(1-\eta_0)\Delta_0 + \sum_{s=0}^{t} \frac{\pi_t}{\pi_s} \frac{\eta_s^2}{2}(2\varepsilon_s/\eta_s + L\left\|\mathbf{a}_s - \mathbf{w}_s\right\|^2). \quad (4.16)$$

To prove the second claim, we have from (4.14)

$$\eta_t \mathsf{G}_t \leq \Delta_t - \Delta_{t+1} + \tfrac{\eta_t^2}{2}(2\varepsilon_t/\eta_t + L\left\|\mathbf{a}_t - \mathbf{w}_t\right\|^2).$$

Summing from $k+1$ to t and noting that we can make $\Delta_{t+1} \geq 0$:

$$\left(\min_{k+1 \leq s \leq t} \mathsf{G}_s \right) \sum_{s=k+1}^{t} \eta_s \leq \sum_{s=k+1}^{t} \eta_s \mathsf{G}_s \leq \Delta_{k+1} + \sum_{s=k+1}^{t} \tfrac{\eta_s^2}{2}(2\varepsilon_s/\eta_s + L\left\|\mathbf{a}_s - \mathbf{w}_s\right\|^2).$$

Rearrange we are done. □

Corollary 4.1. *Under the same setting as in Theorem 4.5, let* $\eta_t = 2/(t+2)$, $\varepsilon_t \leq \delta\eta_t/2$, $L_F := \sup_t L \|\mathbf{a}_t - \mathbf{w}_t\|^2$, *then Algorithm 6 yields*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}) + \frac{2(\delta + L_F)}{t+4}, \tag{4.17}$$

$$\tilde{\mathsf{G}}_t \leq \frac{4.5(\delta + L_F)}{t}. \tag{4.18}$$

Proof. Since $\eta_t = 2/(t+2)$, we have $\eta_0 = 1$ and $\pi_t = \frac{2}{(t+1)(t+2)}$. Thanks to Corollary 4.1, all we need is to verify $\frac{1}{(t+1)(t+2)} \sum_{s=0}^{t} \frac{s+1}{s+2} \leq \frac{1}{t+4}$.

The second claim follows from a sequence of simple (but tedious) calculations. For details, see Freund and Grigas (2013). □

The observation, that the simple step size rule $\eta_t = 2/(t+2)$ already leads to the $O(1/t)$ rate of decrease of the objective value, seems to be due to Clarkson (2010), in the setting where $f = \iota_C$ for some specific compact set C. The similar rate on the minimal duality gap appeared also first in Clarkson (2010) and later extended by Jaggi (2013). The extension to more general f (and more general subroutine `Update`) is straightforward, while the particular case with f strongly convex appeared in Bach (2013b). Of course, it is possible to use other step size rules. For instance, both $\eta_s = 1/(s+1)$ and the constant rule $\eta_s \equiv 1 - \sqrt[t]{t+1}$ lead to an $O(\frac{1+\log t}{t+1})$ rate, see Freund and Grigas (2013) for the detailed calculations. Similar polynomial-decay rules that appeared in Shamir and Zhang (2013) can also be used.

Remark 4.4. *The only catch in Corollary 4.1 is that the "constant"* L_F *might be infinite. Fortunately, we can easily ensure* $L_F < \infty$ *under Proposition 4.1, and there are possibly other ways. We mention again that GCG, with the simple step size rule* $\eta_t = 2/(t+2)$, *does not need to know the Lipschitz constant* L *or specify the norm* $\|\cdot\|$ *(thus one can freely enjoy the "best" setting for his problem). Moreover, the rate in Theorem 4.5 does not depend on the initial point* $\mathbf{w}_0$ *as long as* $\eta_0 = 1$. *On the other hand, by letting* $\eta_0 \neq 1$ *we can optimize the bound which now does depend on how good the initial point* $\mathbf{w}_0$ *is.*

Remark 4.5. *Let us pause and explain the usefulness of the* `Relaxed` *subroutine idea. Consider Algorithm 6 with the "optimal" step size rule (4.10) and with the* `Null` *subroutine; for convenience call this specification Algorithm 007. We have seen in Theorem 4.2 that Algorithm 007 indeed converges asymptotically, but how fast if* ℓ *is convex? Corollary 4.1 above proved the* $O(1/t)$ *rate for Algorithm 6 with step size* $\eta_t = 2/(t+2)$ *and with any* `Relaxed` *subroutine; for convenience call it Algorithm 008. Now realize that the optimal step size rule (4.10), together with the* `Null` *subroutine, consists of nothing but a* `Relaxed` *subroutine for Algorithm 008; and Algorithm 008 with such a* `Relaxed` *subroutine is exactly Algorithm 007. Thus Algorithm 007 enjoys the same* $O(1/t)$ *rate. Note that it is possible to directly prove the rate for Algorithm 007, see for instance Frank and Wolfe (1956); Levitin and Polyak (1966), and Bach (2013b) for a slightly sharper constant. However, we find our argument based on the* `Relaxed` *subroutine idea simpler and cleaner.*

Motivated by our re-interpretation of RDA in Section 1.5, we next show that GCG also converges for the dual problem

$$\inf_{\mathbf{g}} \ell^*(\mathbf{g}) + f^*(-\mathbf{g}). \tag{4.19}$$

Note that when ℓ and f are both closed convex (and subject to some mild regularity conditions) we have from the Fenchel-Rockafellar duality (Zălinescu 2002, Corollary 2.8.5):

$$\inf_{\mathbf{w}} \ell(\mathbf{w}) + f(\mathbf{w}) = -\inf_{\mathbf{g}} \ell^*(\mathbf{g}) + f^*(-\mathbf{g}).$$

The next theorem proves that the averaged gradient $\bar{\mathbf{g}}_T$ automatically solves the dual problem (4.19) at the rate of $O(1/t)$, provided that we can bound the sequences $\{\mathbf{a}_t\}_t, \{\mathbf{w}_t\}_t$ generated in Algorithm 6. This result was first observed in Bach (2013b) by identifying the iterates with those of a modified mirror descent.

Theorem 4.6 (Bach (2013b))**.** *Under Assumption 4.1, and assume that ℓ is convex with L-Lipschitz continuous gradient $\nabla\ell$, that the subroutine is* `Null`*, and that the step size $\eta_t = 2/(t+2)$. Denote $\bar{\mathbf{g}}_{t+1} := \frac{2}{(t+1)(t+2)} \sum_{s=0}^{t} (s+1)\mathbf{g}_s$, then for all $\mathbf{g}$ and $t \geq 1$, Algorithm 6 yields*

$$\ell^*(\bar{\mathbf{g}}_{t+1}) + f^*(-\bar{\mathbf{g}}_{t+1}) \leq \ell^*(\mathbf{g}) + f^*(-\mathbf{g}) + \frac{2}{(t+1)(t+2)} \sum_{s=0}^{t} \frac{s+1}{s+2} L \left\|\mathbf{w}_s + \mathbf{a}_s\right\|^2. \tag{4.20}$$

4.1.4 Positively Homogeneous Case

In this section we consider the special case where f is a positively homogeneous convex function, in short, a gauge. This is motivated by the fact that many regularizers in machine learning are (semi)norms, which are *bona fide* gauges. In particular, our goal is to develop a GCG variant that efficiently solves the matrix completion problem, *cf.* Example 1.10 in Chapter 1.

Before we start, let us point out that GCG is not directly applicable to a gauge function f, simply because the subproblem (4.3) might not have a solution at all. There are two immediate fixes to this. First, we could consider the constrained problem

$$\inf_{\mathbf{w}} \ell(\mathbf{w}) \quad \text{s.t.} \quad f(\mathbf{w}) \leq \zeta. \tag{4.21}$$

It is well-known that (4.21) is equivalent to the regularized problem (4.1), if the constant ζ is chosen appropriately. Moreover, (4.21) usually has a bounded domain therefore GCG can be applied to it. Indeed, a lot of recent works in machine learning are devoted to this variant (4.21), such as (Clarkson 2010; Hazan 2008; Jaggi 2013; Jaggi and Sulovsky 2010; Shalev-Shwartz et al. 2010; Tewari et al. 2011; Yuan and Yan 2013), to name a few. However, (4.21) is a *constrained* problem, hence harder to *locally* improve, due to the need to satisfy the constraint. A second fix is to square f so that it becomes super-coercive (Bradley and Bagnell 2009). When f is a norm, squaring indeed works in finite dimensions. However, it might fail in infinite dimensions (since there not all norms

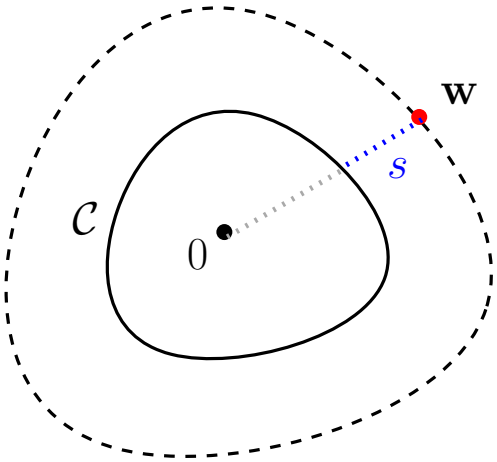


Figure 4.2: The gauge through the Minkowski functional.

are equivalent). Moreover, if we want to insert a local improver in this approach, we would have to evaluate the regularizer f at the iterates. For some applications, such as the matrix completion problem, this is too expensive. A final comment is that if squaring works, how about taking the 3rd power, the 4th power? What is the end of this trick? As it turns out, our proposed variant can be seen as a limit of this process.

As mentioned, we do not want to evaluate the regularizer f at any point, since this might be a very expensive operation. From now on we will switch the notation for the regularizer from f to κ, so that our assumption that κ is a gauge is always signified. It is a well-known fact that a gauge function can be reconstructed from its "unit ball" through the Minkowski functional. Specifically, let $\mathcal{C} := \{\mathbf{w} \in \mathcal{H} : \kappa(\mathbf{w}) \leq 1\}$, then

$$\kappa(\mathbf{w}) = \inf\{\rho : \mathbf{w} \in \rho \cdot \mathcal{C}\}. \tag{4.22}$$

Clearly, $\mathcal{C}$ is a closed and convex set (since κ is assumed to be closed and convex). Intuitively, $\kappa(\mathbf{w})$ is the least amount of stretch (or shrinkage) of the set $\mathcal{C}$ so that it barely touches $\mathbf{w}$, see Figure 4.2. Recall that the *polar* of the gauge κ is defined as

$$\kappa^\circ(\mathbf{g}) := \sup_{\mathbf{w} \in \mathcal{C}} \langle \mathbf{w}, \mathbf{g} \rangle . \tag{4.23}$$

In fact, as shown by Chandrasekaran et al. (2012), one usually starts with a set of "atoms", denoted as $\mathcal{A}$, and construct $\mathcal{C} = \operatorname{conv} \mathcal{A}$, the (closed) convex hull of $\mathcal{A}$. From $\mathcal{C}$ we construct the gauge κ by (4.22). In this case we also have the formula

$$\kappa^\circ(\mathbf{g}) = \sup_{\mathbf{a} \in \mathcal{A}} \langle \mathbf{a}, \mathbf{g} \rangle , \tag{4.24}$$

$$\kappa(\mathbf{w}) = \inf \left\{ \rho : \mathbf{w} = \rho \sum_i \sigma_i \mathbf{a}_i, \sigma_i \geq 0, \sum_i \sigma_i = 1, \mathbf{a}_i \in \mathcal{A} \right\} . \tag{4.25}$$

Algorithm 7 GCG for positively homogeneous regularizers.

Require: The set $\mathcal{A}$ whose (closed) convex hull $\mathcal{C}$ defines the gauge κ.

1: Initialize $\mathbf{w}_0, s_0 \geq \kappa(\mathbf{w}_0)$.
2: **for** $t = 0, 1, \dots$ **do**
3: choose $\mathbf{a}_t$ such that $\langle \mathbf{a}_t, \nabla \ell(\mathbf{w}_t) \rangle \leq \alpha_t \cdot \big(\varepsilon_t + \operatorname{argmin}_{\mathbf{a} \in \mathcal{A}} \langle \mathbf{a}, \nabla \ell(\mathbf{w}_t) \rangle \big)$
4: choose $1 \geq \eta_t \geq 0, \theta_t \geq 0$
5: $\tilde{\mathbf{w}}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \theta_t \mathbf{a}_t$
6: $\tilde{\rho}_{t+1} = (1 - \eta_t)\rho_t + \theta_t$
7: $(\mathbf{w}_{t+1}, \rho_{t+1}) = \texttt{Update}$ ▷ Subroutine, see Theorem 4.7
8: **end for**

Note that we may have infinitely, or even uncountably, many elements in $\mathcal{A}$, but the summation is always well-defined (since $\{\sigma_i\}$ is absolutely summable, it contains at most countably many nonzero entries). At an abstract level, we "lift" the composite problem (4.1) to the infinite dimensional space $\mathfrak{l}_1^\infty$ (the space of absolutely summable sequences), and our convergence corollary below can be seen as a constructive way to prove the denseness of the subspace $\mathfrak{c}_{00}$ (the space of finitely supported sequences) in $\mathfrak{l}_1^\infty$.

The proposed GCG variant is given in Algorithm 7. In each iteration, we find the "atom" $\mathbf{a}_t \in \mathcal{A}$ that (negatively) "correlates" with the gradient of the smooth part ℓ the most (hence yields the rapidest local decrease of ℓ), up to some additive error ε_t and multiplicative factor α_t; then we choose the coefficients η_t and θ_t to (linearly) combine our current iterate $\mathbf{w}_t$ and the "atom" $\mathbf{a}_t$, and similarly for ρ_t; and finally we commit an `Update`, just as before. The scalar variable ρ_t is introduced to handle the unbounded domain: intuitively, ρ_t iteratively estimates $\kappa(\mathbf{w}_t)$ in (4.1). As promised, Algorithm 7 never explicitly evaluates κ; in fact, we need information about κ only in line 3 of Algorithm 7, which essentially computes the polar $\kappa^\circ(-\nabla \ell(\mathbf{w}_t))$. It is evident that Algorithm 7 is a variant of the GCG in Algorithm 6.

We state the main result of this section as follows.

Theorem 4.7. *Under Assumption 4.4, assume that ℓ is convex and $f = h \circ \kappa$ for an increasing finite valued convex function h and a closed gauge κ that is induced by the atomic set $\mathcal{A}$. Choose $\alpha_t > 0$,*

$$0 \leq \eta_t \leq 1, \quad \theta_t = \operatorname*{argmin}_{\theta \geq 0} \theta \langle \mathbf{a}_t, \nabla \ell(\mathbf{w}_t) \rangle + \tfrac{L \|\theta \mathbf{a}_t - \eta_t \mathbf{w}_t\|^2}{2} + \eta_t h(\theta / \eta_t) \tag{4.26}$$

and let the subroutine `Update` be `Relaxed` in the sense that

$$\begin{cases} \ell(\mathbf{w}_{t+1}) + h(\rho_{t+1}) \leq \ell(\mathbf{w}_t) + \langle \theta_t \mathbf{a}_t - \eta_t \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \rangle + \frac{L \|\theta_t \mathbf{a}_t - \eta_t \mathbf{w}_t\|^2}{2} \\ \qquad\qquad + (1 - \eta_t) h(\rho_t) + \eta_t h(\theta_t / \eta_t) \\ \kappa(\mathbf{w}_{t+1}) \leq h(\rho_{t+1}) \end{cases} . \tag{4.27}$$

Then for any $\mathbf{w} \in \operatorname{dom} F$, Algorithm 7 yields

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}) + \pi_t (1 - \eta_0) \big(F(\mathbf{w}_0) - F(\mathbf{w}) \big)$$

$$+\sum_{s=0}^{t}\frac{\pi_t}{\pi_s}\eta_s^2\left(\rho\varepsilon_s+h(\rho/\alpha_s)-h(\rho))/\eta_s+\tfrac{L}{2}\left\|\tfrac{\rho}{\alpha_s}\mathbf{a}_s-\mathbf{w}_s\right\|^2\right), \tag{4.28}$$

where $\rho:=\kappa(\mathbf{w})$, $\pi_t:=\prod_{s=1}^{t}(1-\eta_s)$ *with* $\pi_0=1$.

Proof. Our proof is based upon the following simple observation:

$$F^\star:=\inf_{\mathbf{w}}\{\ell(\mathbf{w})+f(\mathbf{w})\}=\inf_{(\kappa,\rho):\kappa(\mathbf{w})\le\rho}\ell(\mathbf{w})+h(\rho). \tag{4.29}$$

Had we known ρ, we could prove the theorem as before. Intuitively, the step size in (4.26) is chosen to be at least as good as if the algorithm knew the *unknown* but fixed constant $\rho=\kappa(\mathbf{w})$. This is our strategy to prove the theorem.

Note that by construction $\kappa(\mathbf{a}_t)\le 1$. We introduce the scalar estimate ρ_t, which, by construction, is always an upper bound on $\kappa(\mathbf{w}_t)$. We also use the shorthand $\hat{F}_t:=\ell(\mathbf{w}_t)+h(\rho_t)\ge\ell(\mathbf{w}_t)+f(\mathbf{w}_t)=F(\mathbf{w}_t)$.

Let $\rho=\kappa(\mathbf{w})$. The following chain of inequalities is verified:

$$\begin{aligned}
\hat{F}_{t+1} &:= \ell(\mathbf{w}_{t+1})+h(\rho_{t+1})\\
\%\ \texttt{Relaxed}\ \text{subroutine}\ \% &\le \hat{F}_t+\langle\theta_t\mathbf{a}_t-\eta_t\mathbf{w}_t,\nabla\ell(\mathbf{w}_t)\rangle+\tfrac{L}{2}\left\|\theta_t\mathbf{a}_t-\eta_t\mathbf{w}_t\right\|^2-\eta_t h(\rho_t)+\eta_t h(\theta_t/\eta_t)\\
\%\ \text{Optimality of}\ \theta_t\ \% &\le \hat{F}_t+\eta_t\left\langle\tfrac{\rho}{\alpha_t}\mathbf{a}_t-\mathbf{w}_t,\nabla\ell(\mathbf{w}_t)\right\rangle+\tfrac{L}{2}\left\|\tfrac{\rho}{\alpha_t}\mathbf{a}_t-\mathbf{w}_t\right\|^2\eta_t^2-\eta_t h(\rho_t)+\eta_t h(\rho/\alpha_t)\\
\%\ \text{choice of}\ \mathbf{a}_t\ \% &\le \min_{\mathbf{z}:\kappa(\mathbf{z})\le\rho}\hat{F}_t+\eta_t\langle\mathbf{z}-\mathbf{w}_t,\nabla\ell(\mathbf{w}_t)\rangle-\eta_t h(\rho_t)+\eta_t h(\rho)\\
&\qquad+\underbrace{\eta_t^2\left(\tfrac{L}{2}\left\|\tfrac{\rho}{\alpha_t}\mathbf{a}_t-\mathbf{w}_t\right\|^2+(\rho\varepsilon_t+h(\rho/\alpha_t)-h(\rho))/\eta_t\right)}_{:=\delta_t}\\
&= \hat{F}_t+\delta_t-\eta_t\underbrace{\left[\langle\mathbf{w}_t,\nabla\ell(\mathbf{w}_t)\rangle+h(\rho_t)-\min_{\mathbf{z}:\kappa(\mathbf{z})\le\rho}\langle\mathbf{z},\nabla\ell(\mathbf{w}_t)\rangle+h(\rho)\right]}_{:=\hat{\mathsf{G}}(\mathbf{w}_t)}.
\end{aligned}$$

Recall that $\rho=\kappa(\mathbf{w})$, we retrieve the recursion:

$$\begin{aligned}
\hat{F}_{t+1}-F(\mathbf{w})&\le\hat{F}_t-F(\mathbf{w})-\eta_t\hat{\mathsf{G}}(\mathbf{w}_t)+\delta_t,\\
\hat{F}_t-F(\mathbf{w})&\le\hat{\mathsf{G}}(\mathbf{w}_t).
\end{aligned}$$

Expand as in the proof of Theorem 4.5 and note that $F(\mathbf{w}_t)\le\hat{F}_t$ for all t. □

The finite-valued assumption on h is needed to guarantee $h(\rho/\alpha_t)<\infty$. It clearly can be relaxed or even dropped when $\alpha_t\equiv 1$. In particular, with h being the indicator function $\iota_{[0,\zeta]}$ and $\alpha_t\equiv 1$, Corollary 4.2 below implies immediately the same convergence rate for the constrained problem (4.21), recovering (some of) the results discussed in (Clarkson 2010; Hazan 2008; Jaggi 2013; Jaggi and Sulovsky 2010; Shalev-Shwartz et al. 2010; Tewari et al. 2011; Yuan and Yan 2013).

Corollary 4.2. *Under the same setting as in Theorem 4.7, let* $\eta_t=2/(t+2)$, $\varepsilon_t=\delta\eta_t/2$, $\alpha_t\equiv\alpha>0$, $\rho=\kappa(\mathbf{w})$, $L_F:=L\cdot\sup_t\left\|\frac{\rho}{\alpha}\mathbf{a}_t-\mathbf{w}_t\right\|^2$, *then Algorithm 7 yields*

$$F(\mathbf{w}_{t+1})\le F(\mathbf{w})+\frac{2(\rho\delta+L_F)}{t+4}+h(\rho/\alpha)-h(\rho). \tag{4.30}$$

Moreover, if $\ell \geq 0$ *and* $h = \mathsf{Id}$*, then*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w})/\alpha + \frac{2(\rho\delta + L_F)}{t+4}. \tag{4.31}$$

The additional factor ρ in the above bounds is necessary, simply because the inexactness of the subproblem (line 3 of Algorithm 7) is not invariant to scaling hence some compensation is needed. Note also that necessarily we have $\alpha \leq 1$ since the right-hand side of line 3 becomes negative eventually (as $\varepsilon_t \to 0$). The result in (4.31) is very interesting as it roughly says an α-approximate subroutine (for computing the polar of κ) leads to an α-approximate minimizer, again at the rate of $O(1/t)$. We learned later in the defense that Bach (2013a) also considered a similar multiplicative approximation.

Remark 4.6. *Again, we can guarantee* $L_F < \infty$ *under Proposition 4.1. We may also need the atomic set* $\mathcal{A}$ *to be bounded so that* $\mathbf{a}_t$ *in each iteration indeed can be found. Assumption 4.4 is implicitly needed so that* `Relaxed` *subroutine does exist, for instance, the simple rule*

$$\mathbf{w}_{t+1} = \tilde{\mathbf{w}}_{t+1} = (1-\eta_t)\mathbf{w}_t + \theta_t \mathbf{a}_t, \quad \rho_{t+1} = \tilde{\rho}_{t+1} = (1-\eta_t)\rho_t + \theta_t, \tag{4.32}$$

which we call `Null`*. The step size rule for* θ *in (4.26) requires knowledge of the Lipschitz constant* L *and the norm* $\|\cdot\|$*. In particular, if the norm* $\|\cdot\|$ *is Hilbertian and* $h = \mathsf{Id}$*, we have the explicit formula*

$$\theta_t = \left(\frac{\langle \mathbf{a}_t, L\eta_t \mathbf{w}_t - \nabla \ell(\mathbf{w}_t) \rangle - 1}{L \|\mathbf{a}_t\|_{\mathcal{H}}^2} \right)_+. \tag{4.33}$$

It is easy to devise other step size rules that do not require L *or the norm* $\|\cdot\|$*, such as*

$$(\eta_t, \theta_t) \in \operatorname*{argmin}_{1 \geq \eta \geq 0, \theta \geq 0} \ell((1-\eta)\mathbf{w}_t + \theta \mathbf{a}_t) + (1-\eta)h(\rho_t) + \eta h(\theta/\eta), \tag{4.34}$$

followed by taking (4.32). Evidently, (4.34) can be treated as a `Relaxed` *subroutine, hence enjoys the same convergence guarantee in Corollary 4.2.*

Remark 4.7. *Let us compare with the following workaround for solving (4.1). Take an upper bound* $\zeta \geq \kappa(\mathbf{z})$*, where* $\mathbf{z}$ *is our "competitor", say the minimizer of (4.1), and consider*

$$\min_{\mathbf{w}: \kappa(\mathbf{w}) \leq \zeta} \ell(\mathbf{w}) + \kappa(\mathbf{w}). \tag{4.35}$$

We can even dynamically adjust ζ*. Applying GCG in Algorithm 6 requires solving the subproblem*

$$\mathbf{a}_t \in \operatorname*{argmin}_{\mathbf{a}: \kappa(\mathbf{a}) \leq \zeta} \langle \mathbf{a}, \nabla \ell(\mathbf{w}_t) \rangle + \kappa(\mathbf{a}), \tag{4.36}$$

which may be as easy to solve as line 3 in Algorithm 7. However, if we plug in a local improver `Update`*, it is not clear how to maintain the constraint* $\{\mathbf{w} : \kappa(\mathbf{w}) \leq \zeta\}$ *efficiently. Moreover, solving (4.36) up to some* multiplicative *factor might* not *be as easy as line 3 in Algorithm 7. On the other hand, the duality gap* $\mathsf{G}(\mathbf{w})$ *for (4.35) can be computed while* $\hat{\mathsf{G}}(\mathbf{w})$ *(see proof of Theorem 4.7) cannot.*

Surprisingly, there is not much work on the penalized problem (4.1) for a positively homogeneous regularizer. One exception is Dudik et al. (2012), who proposed a totally corrective variant (see (4.37) below). However, their analysis is weak and leads to a suboptimal $O(1/\sqrt{t})$ rate of convergence.

4.1.5 Refinements and Comments

We briefly mention some possible refinements and further comments in this section.

So far, to derive concrete rates of convergence in the convex (and positively homogeneous) case, we have assumed that the loss ℓ has Lipschitz continuous gradient. This assumption, although holds for a variety of losses such as the square loss and the logistic loss, does fail particularly for the hinge loss in SVM (*cf.* Example 1.2). However, we can always first "smooth" the nonsmooth loss using the Moreau envelop discussed in Chapter 3, and then apply GCG. This usually results in a slower $O(1/\sqrt{t})$ rate of convergence, though.

Several of our proofs rely on the particular step size rule $\eta_t = O(\frac{1}{t})$, which appears to be "optimal", among non-adaptive ones, in the following sense. On the one hand, we usually prefer large step sizes since they often result in faster convergence; on the other hand, the algorithm needs to be able to remove some atom $\mathbf{a}_t$ that is perhaps "incidentally" added. This requires the discount factor $\prod_{t=1}^{\infty}(1-\eta_t)$ to be as small as needed. It is an easy exercise to prove that the latter condition holds if and only if $\sum_{t=1}^{\infty}\eta_t = \infty$. Therefore the step size rule $O(\frac{1}{t})$ is (almost) the largest non-adaptive one that still allows removing some "atom" (which might be crucial for convergence).

Algorithm 7 amounts to adding one more "atom" in each iteration, followed by balancing the old atoms, as a whole, and the new atom. An even more aggressive scheme is to completely re-optimize the weights of all atoms in each iteration. This procedure was first studied by Meyer (1974) and is generally known as the totally (sometimes referred to as fully) corrective update in the boosting literature. Mathematically, in each iteration we solve (for the positively homogeneous case with $h = \mathsf{Id}$)

$$\min_{\boldsymbol{\sigma}\geq 0} \ell\left(\sum_{\tau=1}^{t}\sigma_\tau \mathbf{a}_\tau\right) + \sum_{\tau=1}^{t}\sigma_\tau. \tag{4.37}$$

Not surprisingly, the totally corrective variant can be seen as a `Relaxed` subroutine in Algorithm 7, hence converges at least as fast as Corollary 4.2 suggests. Empirically, much faster convergence is usually observed, although this advantage must be countered by the extra effort spent in solving (4.37), which itself need not be trivial at all. In a finite dimensional setting, provided that the atoms are linearly independent and some restricted strong convexity is present, it is possible to prove that the totally corrective variant (4.37) converges at a linear rate (ignoring the per-step complexity), see Shalev-Shwartz et al. (2010); Yuan and Yan (2013).

Lastly, we remark that the derived convergence rate of GCG is on par with that of PG (*cf.* Section 1.3) and cannot be improved even in the presence of strong convexity (Canon and Cullum 1968).

Thus GCG is slower than the "optimal" algorithm APG. The potential gain is that GCG only needs to solve a linear subproblem (*i.e.*, polar of the gauge in the positively homogeneous case) in each iteration, while APG (or PG) requires computing the proximal map which is a quadratic problem. The two algorithms seem to complement each other since in some cases the polar is easier to compute while we saw before in some other cases the proximal map can be computed analytically. Another advantage of GCG over APG lies in its greedy nature: Each iteration of GCG amounts to adding one more atom, therefore the total number of atoms, namely a meaningful form of sparsity, does not exceed the number of iterations that GCG takes. In contrast, APG might yield dense estimates in one iteration, although in later stages the estimates may become sparser due to the shrinkage effect of the proximal map. More importantly, GCG is "robust" with respect to an α-approximate polar subroutine (*cf.* (4.31) in Corollary 4.2) while we are not aware of a similar result for PG or APG with respect to the proximal map.

4.1.6 Examples

In this section we discuss some salient examples of the GCG algorithm.

Let us first check the matrix completion example in Example 1.10, with the newly developed GCG variant in Algorithm 7.

Example 1.10 (continuing from p. 18)**.** *We have seen previously in Example 1.10 that the proximal map of the trace norm, as needed in PG or APG, has an analytic form, but it requires a* full *SVD on the iterate, which can be prohibitively expensive in large-scale applications. On the other hand, line 4 of Algorithm 7 (the only nontrivial step) amounts to computing the polar of the trace norm, which is simply the spectral norm, an order of magnitude cheaper than a full SVD (more specifically, $O(n^3)$ versus $O(n^2)$ for an $n \times n$ matrix). Thus, even though GCG has a slower theoretical rate than APG, its per-step complexity can be much cheaper in matrix applications. Overall, it still seems preferable to use GCG rather than APG, as shown in our experiments below.*

The next example demonstrates that PG can be regarded as a special case of GCG.

Example 4.1 (PG $\subset$ GCG)**.** *This is the main motivation of Bredies et al. (2009) to study GCG. Recall that we never assumed the convexity of ℓ until we started to derive concrete convergence rates in Section 4.1.3 and Section 4.1.4. For the composite problem $\inf_{\mathbf{w}} \ell(\mathbf{w}) + f(\mathbf{w})$, PG upper bounds the smooth loss by some quadratic and solves, in each iteration, the proximal map*

$$\inf_{\mathbf{z}} \ell(\mathbf{w}) + \langle \mathbf{z} - \mathbf{w}, \nabla \ell(\mathbf{w}) \rangle + L \mathsf{D}_{\mathsf{d}}(\mathbf{z}, \mathbf{w}) + f(\mathbf{w}), \tag{4.38}$$

where L is the Lipschitz constant of the gradient $\nabla \ell$ and $\mathsf{D}_{\mathsf{d}}(\mathbf{z}, \mathbf{w}) := \mathsf{d}(\mathbf{z}) - \mathsf{d}(\mathbf{w}) - \langle \mathbf{z} - \mathbf{w}, \nabla \mathsf{d}(\mathbf{w}) \rangle$ is the Bregman divergence induced by d*. Now consider the equivalent problem*

$$\inf_{\mathbf{w}} \underbrace{\ell(\mathbf{w}) - L\mathsf{d}(\mathbf{w})}_{\tilde{\ell}(\mathbf{w})} + \underbrace{L\mathsf{d}(\mathbf{w}) + f(\mathbf{w})}_{\tilde{f}(\mathbf{w})}. \tag{4.39}$$

When d *has 1-Lipschitz continuous gradient,* $\tilde{\ell}$ *has* $2L$*-Lipschitz continuous gradient*[6]*, therefore we can linearize* $\tilde{\ell}$ *and apply GCG, which, in each iteration, solves*

$$\inf_{\mathbf{a}} \langle \mathbf{a}, \nabla \ell(\mathbf{w}) - L\nabla \mathsf{d}(\mathbf{w}) \rangle + L\mathsf{d}(\mathbf{a}) + f(\mathbf{a}). \tag{4.40}$$

Clearly, (4.38) *and* (4.40) *are equivalent, thus PG can be seen as a special case of GCG. On the other hand, it does not seem possible to reduce GCG to PG (without violating the assumption that* f *is convex). Somewhat disappointingly, this reduction from GCG to PG does not appear to be very useful in our opinion, for instance, we can not prove the* $O(1/t)$ *convergence rate for GCG under the assumption that* f *and* $\ell + f$ *are convex; we needed both* ℓ *and* f *to be. Had we been able to prove the rate under the former condition, we could recover the* $O(1/t)$ *of PG from that of GCG, which would be another interesting result.*

Our last example is about the celebrated Adaboost algorithm.

Example 4.2 (Adaboost $\subset$ GCG)**.** *The celebrated Adaboost of Freund and Schapire (1997) is another instance of GCG. In fact, our development of the GCG variant for positively homogeneous regularizers was motivated by the desire to add regularization to boosting.*

Let us first recall the Adaboost algorithm, in the setting of binary classification, cf. *Example 1.1. Given a training sample* $(\mathbf{x}_i, y_i)_{i=1}^n$ *where say* $\mathbf{x}_i \in \mathbb{R}^m, y_i \in \{1, -1\}$*, and a set of "weak" classifiers* $h_j : \mathbb{R}^m \to \{-1, 1\}$ *for* $j \in J$*, our goal is to find a weight vector* $\mathbf{w}$ *such that the linear combination* $\sum_j w_j h_j$ *minimizes the classification error,* i.e.*, a "strong" classifier. We have seen in Example 1.1 that directly minimizing the 0-1 loss on the training set is hard, so we turn to the exponential loss, which is a convex upper bound:*

$$\min_{\mathbf{w}\in\mathbb{R}^J} \tfrac{1}{n} \sum_{i=1}^n \exp(-y_i F(\mathbf{w})), \quad \textit{where} \quad F(\mathbf{w}) = \sum_j w_j h_j(\mathbf{x}_i). \tag{4.41}$$

Clearly the exponential loss is continuously differentiable, however, (4.41) *has an unbounded domain* $\mathbb{R}^J$*, therefore we cannot apply the usual GCG. But the GCG variant for positively homogeneous regularizers is still applicable: simply pretend that we have a regularizer, say* $0 \cdot \|\mathbf{w}\|_1$*. Then the first step in Algorithm 7 becomes (at iteration* t*):*

$$h_{t+1} \in \operatorname*{argmax}_{h\in\{h_j : j\in J\}} \left| \tfrac{1}{n} \sum_{i=1}^n \exp(-y_i F(\mathbf{w}_t)) \cdot y_i h(\mathbf{x}_i) \right|, \tag{4.42}$$

i.e.*, selecting a new weak classifier. Next we choose the step size* η_t *and* θ_t*. Due to the inherent homogeneity of the objective in* (4.41)*, we have*

$$\exp\Big(- y_i \Big(\theta h_{t+1}(\mathbf{x}_i) + \sum_{j=1}^t (1-\eta) w_j h_j(\mathbf{x}_i) \Big) \Big) \propto \exp\Big(- y_i \Big(\tilde{\theta} h_{t+1}(\mathbf{x}_i) + \sum_{j=1}^t w_j h_j(\mathbf{x}_i) \Big) \Big),$$

[6] The constant can be reduced to L if ℓ is convex, everywhere defined and $\mathsf{d} = \frac{1}{2}\|\cdot\|_{\mathcal{H}}^2$.

therefore w.l.o.g. we let $\eta_t = 0$ *and find* $\tilde{\theta}$ *by the line search*

$$\min_{\theta \geq 0} \sum_{i=1}^{n} \exp\Big(- y_i \Big(\theta h_{t+1}(\mathbf{x}_i) + \sum_{j=1}^{t} w_j h_j(\mathbf{x}_i) \Big) \Big),$$

which leads to the analytic solution

$$\tilde{\theta}_t = \left(\tfrac{1}{2} \ln \frac{\sum_{i:h_{t+1}(\mathbf{x}_i)=y_i} \exp(-y_i \sum_{j=1}^{t} w_j h_j(\mathbf{x}_i))}{\sum_{i:h_{t+1}(\mathbf{x}_i)\neq y_i} \exp(-y_i \sum_{j=1}^{t} w_j h_j(\mathbf{x}_i))} \right)_+ . \tag{4.43}$$

We have thus recovered precisely the Adaboost algorithm, which (computationally speaking) is merely our Algorithm 7 with an analytic line search. Of course one could equally well "pretend" that we have a different regularizer other than $0 \cdot \|\mathbf{w}\|_1$*; the advantage of the latter is that it leads to the greedy coordinate-wise step (4.42). The viewpoint to think of Adaboost as some greedy algorithm is well-known, see* e.g. *Mason et al. (2000); Zhang (2003).*

A popular theory to explain Adaboost's empirical success is its margin *maximization. However, Grove and Schuurmans (1998) designed the LPBoost to explicitly maximize the margin but observed severe overfitting. Warmuth et al. (2008) then considered the entropy regularized LPBoost, which is further extended in Shalev-Shwartz and Singer (2010). Their algorithms are again straightforward instances of GCG, applied to the* smoothed *loss of LPBoost.*

Of course, there are many other examples of GCG, after all it is such a simple yet effective algorithm. For some applications in nonlinear function approximation, see Temlyakov (2011). Another closely related example is the matching pursuit algorithm in signal processing, see Mallat (2009).

4.2 Dictionary Learning

We have briefly mentioned the dictionary learning problem in Example 1.5 of Chapter 1. It turns out that the GCG algorithm we developed in the previous section suits the needs of dictionary learning very well, and the current section is devoted to demonstrating this point.

4.2.1 Convex Relaxation

To begin with, let us recall the dictionary learning problem (Olshausen and Field 1996). We are given an $n \times m$ matrix X, each column of which corresponds to a training example and the rows represent different features across examples. Our goal is to learn an $n \times k$ "dictionary" matrix U, consisting of k basis vectors, and a $k \times m$ coefficient matrix V, such that UV approximates X in the sense of minimizing some loss $\ell(UV, X) = \ell(UV)$. The problem is not well-defined yet since we can always scale the matrix U up and scale the matrix V down accordingly, without changing their product UV. Therefore, to remove this scaling invariance, it is customary to restrict the bases, *i.e.* columns of U, to the unit ball of some norm $\|\cdot\|_c$ (c for column). There can be other constraints on U or V.

The key of dictionary learning is to learn both the dictionary and the coefficients *simultaneously*. This is in sharp contrast with traditional signal approximation schemes where one fixes the dictionary, say the Fourier basis or some wavelet basis, *a priori*. Unfortunately, the added flexibility of dictionary learning also brings much computational challenge, as the formulation is no longer *jointly* convex in the variables U and V, even when the loss ℓ is convex (in its first argument). Indeed, for a fixed dictionary size k, the dictionary learning problem is known to be computationally tractable only for losses induced by unitarily invariant norms (Yu and Schuurmans 2011). With nonnegative constraints on U and V, namely the nonnegative matrix factorization of Lee and Seung (1999), the problem is NP-Hard even for the squared loss (Vavasis 2010).[7] To retain tractability for a variety of convex losses, a popular and successful approach is to consider "relaxations" that avoid the "hard" constraint on the size of the dictionary, *i.e.* a fixed k. As a compensation we can add an appropriate regularizer on the magnitude of rows of the coefficient matrix V so that overall dictionaries with a small size are still encouraged. A second motivation to "relax" k arises from the fact that we usually do not know k beforehand; why not let the algorithm decide the "best" one for us?

Specifically, the following relaxation has been considered by a number of people, *e.g.* Argyriou et al. (2008); Bach et al. (2008); Bradley and Bagnell (2009); Zhang et al. (2011):

$$\inf_{U:\|U_{:i}\|_c \le 1} \inf_{\tilde{V}} \ell(U\tilde{V}) + \lambda \sum_i \|\tilde{V}_{i:}\|_r, \tag{4.44}$$

where $\lambda \ge 0$ balances the trade-off between the loss and the regularizer, and $U_{:i}, \tilde{V}_{i:}$ denote the i-th column and row of U and $\tilde{V}$, respectively. The idea, as shown in Figure 4.3, is that by minimizing the regularized problem (4.44), many rows of $\tilde{V}$ will become exactly zero due to the row-norm regularizer, therefore accordingly the corresponding columns of U can be dropped, resulting in a small dictionary. Moreover, the specific form of the row norm $\|\cdot\|_r$ provides additional flexibility in promoting different structures, such as: the ℓ_1 norm leads to sparse solutions; the ℓ_2 norm yields low rank solutions; and block structured norms generate group sparsity. The specific form of the column norm $\|\cdot\|_c$ also has an effect, see Example 4.3 below.

The fact that (4.44) can be reformulated as a convex problem was first realized in Bach et al. (2008) and later rediscovered in Zhang et al. (2011). Following Zhang et al. (2012), we present a concise proof of this general observation, through the use of gauge functions. First, we do a normalization so that $\tilde{V}_{i:} = \sigma_i V_{i:}$, where $\sigma_i \ge 0$ and $\|V_{i:}\|_r \le 1$. Now (4.44) can be reformulated by introducing the reconstruction matrix $W := U\tilde{V}$:

$$\begin{aligned}(4.44) &= \min_W \ell(W) + \lambda \cdot \inf \left\{ \sum_i \left\|\tilde{V}_{i:}\right\|_r : \|U_{:i}\|_c \le 1, U\tilde{V} = W \right\} \\ &= \min_W \ell(W) + \lambda \cdot \inf \left\{ \sum_i \sigma_i : \boldsymbol{\sigma} \ge 0, W = \sum_i \sigma_i U_{:i} V_{i:}, \|U_{:i}\|_c \le 1, \|V_{i:}\|_r \le 1 \right\}\end{aligned}$$

[7] Some recent work (Arora et al. 2012; Recht et al. 2012) has shown the tractability of this model under some separability assumptions.

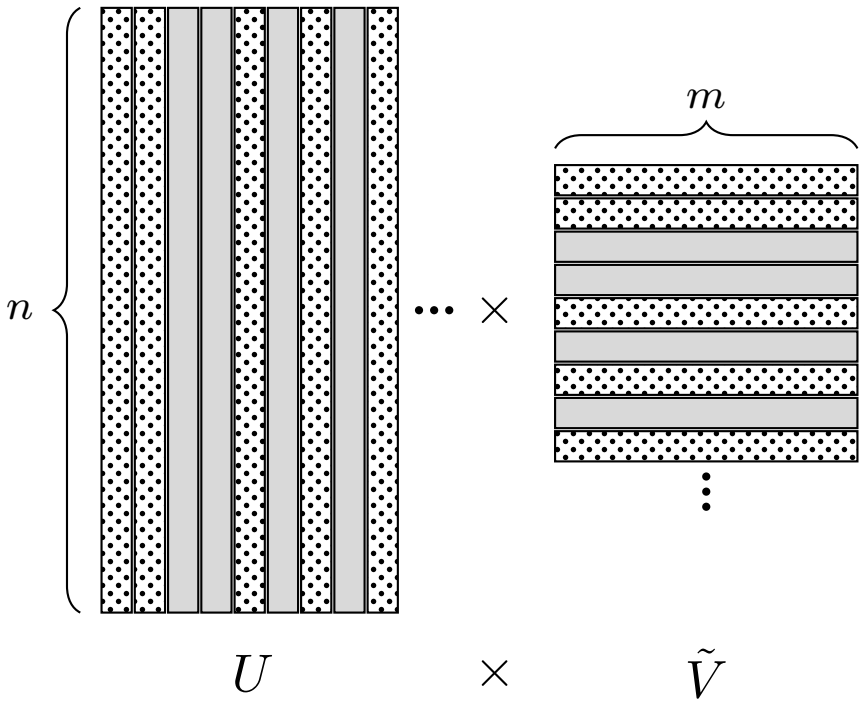


Figure 4.3: The idea behind the convex relaxation for dictionary learning: Due to the row-wise norm regularizer on $\tilde{V}$, many rows (grayed) of it will become exactly zero, therefore the corresponding columns of U will be dropped, resulting in a small dictionary.

$$= \min_{W} \ell(W) + \lambda \cdot \kappa(W), \tag{4.45}$$

where the gauge κ is induced by the set

$$\mathcal{A} := \{\mathbf{u}\mathbf{v}^\top : \mathbf{u} \in \mathbb{R}^n, \mathbf{v} \in \mathbb{R}^m, \|\mathbf{u}\|_c \leq 1, \|\mathbf{v}\|_r \leq 1\} \tag{4.46}$$

through the construction (4.25). Note that the set $\mathcal{A}$ contains uncountably many elements. Clearly (4.45) is a convex problem, provided that the loss ℓ is convex (in its first argument).

The reformulation (4.45) is illuminating in a few aspects. Firstly, it reveals that the regularized dictionary learning problem (4.44) is nothing but a rank-one decomposition of the matrix X under the loss ℓ, penalized by the sum of "singular" values $\{\sigma_i\}$, a proxy of the "rank" or dictionary size k. Observe the striking similarity with the singular value decomposition. Secondly, we now have a better understanding of what the added regularizer in (4.44) is really doing. This is achieved by checking the polar

$$\begin{aligned} \kappa^\circ(Z) &= \sup_{A \in \mathcal{A}} \langle A, Z \rangle = \sup_{\|\mathbf{u}\|_c \leq 1, \|\mathbf{v}\|_r \leq 1} \mathbf{u}^\top Z \mathbf{v} \\ &= \sup_{\|\mathbf{u}\|_c \leq 1} \left\| Z^\top \mathbf{u} \right\|_r^\circ \qquad (4.47) \\ &= \sup_{\|\mathbf{v}\|_r \leq 1} \|Z\mathbf{v}\|_c^\circ, \qquad (4.48) \end{aligned}$$

where for convenience we have changed the notation for the dual norm from $\|\cdot\|_\circ$ to $\|\cdot\|^\circ$. This should not cause any confusion. In other words, the polar κ° is simply the induced matrix norm. Using the duality $\kappa = (\kappa^\circ)^\circ$ we can get an explicit formula for the gauge regularizer κ as well. For example, if $\|\cdot\|_r = \|\cdot\|_c = \|\cdot\|_2$, we have $\kappa^\circ(\cdot) = \|\cdot\|_{\mathrm{sp}}$ (the spectral norm), and the regularizer in this case is $\kappa(\cdot) = \|\cdot\|_{\mathrm{tr}}$ (the trace norm). Thus the trace norm regularization we saw in the matrix completion example (*cf.* Example 1.10) can be explained from the point of view of dictionary learning. Note that some care on choosing the norms $\|\cdot\|_c$ and $\|\cdot\|_r$ is needed, since otherwise we might get trivial results, see Zhang et al. (2011). In particular, we have the next interesting result.

Example 4.3. *Sometimes, our data may be naturally divided into different categories. For instance, in semi-supervised learning (Zhang et al. 2011) we have both the labeled and unlabeled training data while in multi-view learning (White et al. 2012) we have data from different "views" of the same object. As an example, consider the case where each column* $\mathbf{x}$ *of* X *is formed by two subvectors* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*. Correspondingly we subdivide each basis vector* $\mathbf{u}$ *in the dictionary* U *into two subvectors* $\mathbf{u}_1$ *and* $\mathbf{u}_2$*. Following the above recipe we arrive at the "atomic" set* $\mathcal{A}$ *in* (4.46)*. Which column norm* $\|\cdot\|_c$ *should we use? Instead of using* $\|\mathbf{u}\|_2$ *and letting* $\mathbf{u}_1$ *and* $\mathbf{u}_2$ *compete against each other, it seem to make more sense to use the norm* $\max\{\|\mathbf{u}_1\|_2, \|\mathbf{u}_2\|_2\}$*, to "separate"* $\mathbf{u}_1$ *and* $\mathbf{u}_2$*. What is the resulting induced norm? Let us take* $\|\cdot\|_r = \|\cdot\|_2$ *and check the polar first:*

$$\begin{aligned}
\kappa_\circ^2(Z) &= \sup\left\{\mathbf{u}^\top ZZ^\top \mathbf{u} : \|\mathbf{u}_1\|_2 \le 1, \|\mathbf{u}_2\|_2 \le 1\right\} \\
&= \sup\left\{\mathrm{tr}(SZZ^\top) : \mathrm{tr}(SI_1) \le 1, \mathrm{tr}(SI_2) \le 1, S \succeq 0\right\} \\
&= \sup_{S\succeq 0} \inf_{\mu\ge 0,\nu\ge 0} \mathrm{tr}(SZZ^\top) - \mu(\mathrm{tr}(SI_1) - 1) - \nu(\mathrm{tr}(SI_2) - 1) \\
&= \inf_{\mu\ge 0,\nu\ge 0} \sup_{S\succeq 0} \mathrm{tr}(SZZ^\top) - \mu(\mathrm{tr}(SI_1) - 1) - \nu(\mathrm{tr}(SI_2) - 1) \\
&= \inf\left\{\mu + \nu : \mu \ge 0, \nu \ge 0, ZZ^\top \preceq \mu I_1 + \nu I_2\right\} \\
&= \inf\left\{\mu + \nu : \mu \ge 0, \nu \ge 0, \left\|D_{\nu/\mu} Z\right\|_{\mathrm{sp}}^2 \le \mu + \nu\right\} \\
&= \inf\left\{\|D_\rho Z\|_{\mathrm{sp}}^2 : \rho \ge 0\right\},
\end{aligned}$$

where $D_\rho = \mathrm{diag}(\sqrt{1+\rho} I_1, \sqrt{1+1/\rho} I_2)$ *is a diagonal scaling of the identity matrix* I_1 *(on the subspace spanned by* $\mathbf{u}_1$*) and* I_2 *(on the subspace spanned by* $\mathbf{u}_2$*), and the second equality is obtained from dropping the rank-1 constraint on* $S = \mathbf{u}\mathbf{u}^\top$*.*[8] *Therefore the (squared) polar is simply the infimum of a family of re-scaled (squared) spectral norms; a duality argument then shows that the (squared) gauge is the supremum of a family of re-scaled (squared) trace norms. After a re-parameterization, White et al. (2012) further proves that this gauge is concave in the parameter* ρ*, which then allows them to efficiently solve the resulting convex-concave program. However, a simpler approach would be to directly apply GCG, which is discussed in Zhang et al. (2012) under a more complicated setting.*

[8] Since we only have two linear inequalities, dropping the rank constraint does not increase the objective, simply because the maximum of a linear function over a convex set is attained at one of the extreme points, whose rank can be upper bounded by 1.

The above convex relaxation framework, which is based on gauge functions, is quite flexible and has been studied in a number of structured sparse problems (Chandrasekaran et al. 2012; Tewari et al. 2011). Computationally, our GCG variant in Algorithm 7 is a very natural candidate for optimizing the resulting convex problem (4.45), as in each iteration we need only compute the polar through either (4.47) or (4.48). However, this simplicity must be countered by the fact that the induced norm is not always tractable, after all we are *maximizing* a norm subject to a different norm constraint. This is our main motivation to introduce an α-approximate polar oracle in Algorithm 7 and Corollary 4.2.

4.2.2 Fixed-Rank Local Optimization

As mentioned, our GCG variant in Algorithm 7 can be readily applied to the reformulated dictionary learning problem (4.45), however, the sublinear rate of convergence established in Corollary 4.2 is still too slow in large-scale applications. By exploiting the matrix structure, we present in this section a simple acceleration trick which can be regarded as a `Relaxed` subroutine in Algorithm 7.

Recall that in the `Null` version of Algorithm 7 (*cf.* (4.32)), $\mathbf{w}_{t+1}$ is determined by some linear combination of the previous iterate $\mathbf{w}_t$ and the newly added atom $\mathbf{a}_t$. We first demonstrate that we can further improve $\mathbf{w}_{t+1}$ by solving some related but different surrogate problem. Next, we address the issue of restoring the "context" for Algorithm 7. Two simple propositions turn out to be the key.

Proposition 4.2. *The gauge κ induced by the set $\mathcal{A}$ in (4.46) can be re-expressed as*

$$\kappa(W) = \inf\left\{ \tfrac{1}{2}\sum_i \left(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2 \right) : UV = W \right\} \tag{4.49}$$

$$= \inf\left\{ \sum_i \|U_{:i}\|_c \cdot \|V_{i:}\|_r : UV = W \right\}. \tag{4.50}$$

Proof. The proof is similar in spirit to that of Bach et al. (2008). For any $UV = W$, we have the normalization

$$W = \sum_i \|U_{:i}\|_c \|V_{i:}\|_r \tfrac{U_{:i}}{\|U_{:i}\|_c} \tfrac{V_{i:}}{\|V_{i:}\|_r}.$$

Thus by the definition of the gauge κ in (4.45),

$$\kappa(W) \le \sum_i \|U_{:i}\|_c \|V_{i:}\|_r \le \tfrac{1}{2}\sum_i \left(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2 \right).$$

On the other hand, for any $\epsilon > 0$, there exist $\boldsymbol{\sigma} \ge 0$, $\hat{U}$, and $\hat{V}$ such that

$$\forall i, \|\hat{U}_{:i}\|_c = \|\hat{V}_{i:}\|_r = 1; \qquad \sum_i \sigma_i \hat{U}_{:i}\hat{V}_{i:} = W; \qquad \kappa(W) + \epsilon \ge \sum_i \sigma_i.$$

Define $U_{:i} = \sqrt{\sigma_i}\hat{U}_{:i}$ and $V_{i:} = \sqrt{\sigma_i}\hat{V}_{i:}$. We verify that $UV = W$ and

$$\tfrac{1}{2}\sum_i (\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2) = \sum_i \|U_{:i}\|_c \|V_{i:}\|_r = \sum_i \sigma_i \le \kappa(W) + \epsilon.$$

Since $\epsilon > 0$ is arbitrary, taking limits we obtain the claim in the proposition. □

As mentioned, if $\|\cdot\|_r = \|\cdot\|_c = \|\cdot\|_2$, κ reduces to the trace norm, and the term $\sum_i(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2)$ is simply $\|U\|_F^2 + \|V\|_F^2$, the sum of the (squared) Frobenius norms. In this case Proposition 4.2 is a well-known variational form of the trace norm (Srebro et al. 2005). This motivates us to choose the auxiliary function

$$\mathfrak{F}_t(U, V) := \ell(UV) + \tfrac{\lambda}{2} \sum_{i=1}^{t} \Big(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2 \Big), \tag{4.51}$$

which can be locally optimized to accelerate the overall convergence. Note that (4.51), as in (4.44), is not *jointly* convex. Moreover, the difference between (4.51) and the original dictionary learning problem (4.44) is that we have fixed the size t (the iteration counter) in (4.51). As can be imagined, if t is sufficiently large and the initialization to (4.51) is good enough, any local minimizer of (4.51) is often acceptable for the original problem (4.44); see Burer and Monteiro (2005) for some formal justification.

The advantage of the surrogate objective in (4.51) is that it is usually smooth, and the regularizer is separable in U and V. Moreover, we only aim at a locally improved solution, therefore can "solve" (4.51) rather quickly. The subtlety is how to switch back to the GCG variant, without ruining its convergence property, after all the surrogate (4.51) is different from our initial problem (4.45). The next proposition ensures that we can recover (if we want) the atoms from any local minimizer of $\mathfrak{F}_t$.

Proposition 4.3. *For any $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$, there exist $\sigma_i \geq 0$, $\mathbf{u}_i \in \mathbb{R}^m$, and $\mathbf{v}_i \in \mathbb{R}^n$ such that*

$$UV = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \quad \|\mathbf{u}_i\|_c \leq 1, \quad \|\mathbf{v}_i\|_r \leq 1, \quad \sum_{i=1}^{k} \sigma_i = \tfrac{1}{2} \sum_{i=1}^{k} \Big(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2 \Big).$$

Proof. Denote $a_i = \|U_{:i}\|_c$ and $b_i = \|V_{i:}\|_r$. Then

$$UV = \sum_i a_i b_i \frac{U_{:i}}{a_i} \frac{V_{i:}}{b_i} = \sum_i \underbrace{\tfrac{1}{2}(a_i^2 + b_i^2)}_{:=\sigma_i} \underbrace{\sqrt{\frac{a_i b_i}{\frac{1}{2}(a_i^2 + b_i^2)}} \frac{U_{:i}}{a_i}}_{:=\mathbf{u}_i} \underbrace{\sqrt{\frac{a_i b_i}{\frac{1}{2}(a_i^2 + b_i^2)}} \frac{V_{i:}}{b_i}}_{:=\mathbf{v}_i^\top}.$$

Clearly $\|\mathbf{u}_i\|_c \leq 1$, $\|\mathbf{v}_i\|_r \leq 1$, and $\sum_i \sigma_i = \frac{1}{2}\sum_i(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2)$. □

In fact, all we need for the next iteration of our GCG variant in Algorithm 7 is W_{t+1} and ρ_{t+1}, which can be computed *directly* from any local minimizer (U^*, V^*) of (4.51), hence keeping the recovery (of the atoms) completely implicit:

$$W_{t+1} = U^* V^* \quad \text{and} \quad \rho_{t+1} = \tfrac{1}{2} \sum_{i=1}^{t+1} \Big(\|U_{:i}^*\|_c^2 + \|V_{i:}^*\|_r^2 \Big). \tag{4.52}$$

In addition, Proposition 4.3 ensures that improving $\mathbf{w}_{t+1}$ through locally minimizing the surrogate (4.51) does not incur an increase in the number of atoms.

The final algorithm is summarized in Algorithm 8. The first two steps are the same as in Algorithm 7. Line 5 carefully splits the iterate into two parts U_{init} and V_{init}, while line 6 finds a local

Algorithm 8 GCG variant for dictionary learning.

Require: The atomic set $\mathcal{A}$.

Initialize $W_0 = \mathbf{0}$, $s_0 = 0$, $U_0 = V_0 = \Lambda_0 = \emptyset$.
for $t = 0, 1, \dots$ **do**
  $(\mathbf{u}_t, \mathbf{v}_t) \leftarrow \operatorname*{argmin}_{\mathbf{u}\mathbf{v}^\top \in \mathcal{A}} \langle \nabla \ell(W_t), \mathbf{u}\mathbf{v}^\top \rangle$
  $(\eta_t, \theta_t) \leftarrow \operatorname*{argmin}_{0 \le \eta \le 1, \theta \ge 0} \ell((1-\eta)W_t + \theta\, \mathbf{u}_t\mathbf{v}_t^\top) + \lambda((1-\eta)\rho_t + \theta)$
  $U_{\text{init}} \leftarrow (\sqrt{1-\eta_t}U_t, \sqrt{\theta_t}\mathbf{u}_t)$, $V_{\text{init}} \leftarrow (\sqrt{1-\eta_t}V_t, \sqrt{\theta_t}\mathbf{v}_t)^\top$
  $(U_{t+1}, V_{t+1}) = \texttt{Update}(\mathfrak{F}_{t+1}, U_{\text{init}}, V_{\text{init}})$
  $W_{t+1} \leftarrow U_{t+1}V_{t+1}$
  $\rho_{t+1} \leftarrow \frac{1}{2}\sum_{i=1}^{t+1}(\|(U_{t+1})_{:i}\|_c^2 + \|(V_{t+1})_{i:}\|_r^2)$
end for

minimizer of the surrogate (4.51), with the designated initialization. The last two steps restore the iterate of GCG. To see that Algorithm 8 still enjoys the $O(1/t)$ rate of convergence established in Corollary 4.2, it is enough to prove that the introduced `Update` subroutine is `Relaxed`. Indeed, by construction

$$\begin{aligned}
\ell(W_{t+1}) + \lambda\rho_{t+1} &= \ell(U_{t+1}V_{t+1}) + \tfrac{\lambda}{2}\sum_{i=1}^{t+1}\|(U_{t+1})_{:i}\|_c^2 + \|(V_{t+1})_{i:}\|_r^2 \\
&= \mathfrak{F}_{t+1}(U_{t+1}, V_{t+1}) \\
&\le \mathfrak{F}_{t+1}(U_{\text{init}}, V_{\text{init}}) \\
&\le \ell((1-\eta_t)U_tV_t + \theta_t\mathbf{u}_t\mathbf{v}_t^\top) + \lambda\theta_t + \tfrac{\lambda(1-\eta_t)}{2}\sum_{i=1}^{t}\|(U_t)_{:i}\|_c^2 + \|(V_t)_{i:}\|_r^2 \\
&= \ell((1-\eta_t)W_t + \theta_t\mathbf{u}_t\mathbf{v}_t^\top) + \lambda((1-\eta_t)\rho_t + \theta_t),
\end{aligned}$$

and of course $\kappa(W_{t+1}) \le \rho_{t+1}$, thanks to Proposition 4.2. Although we were not able to prove any *strict* improvement brought by the local subroutine, we observed in the experiments that Algorithm 8 is usually much faster than the `Null` version of Algorithm 7. In other words, local acceleration seems to make a big difference in practice.

Interlacing local improvement with some globally convergent procedure itself is not a new idea. Closely related to our proposal is the work of Mishra et al. (2013) and Laue (2012). Laue (2012) considered the constrained problem (4.21), therefore his local procedure is also a constrained problem, which might be less efficient than our unconstrained surrogate (4.51). Targeting specifically at the trace norm regularizer, Mishra et al. (2013) proposed a trust-region procedure to locally optimize the *original* objective on the Stiefel manifold and the positive semidefinite cone. They need to dynamically maintain the singular value decomposition of a small matrix and their local procedure is also performed on a constrained problem. Furthermore, no rate of convergence is established in Mishra et al. (2013).

4.3 Experimental Results

We compare the GCG variant in Algorithm 8 to three state-of-the-art solvers, MMBS[9] (Mishra et al. 2013), DHM (Dudik et al. 2012), and JS (Jaggi and Sulovsky 2010), for the trace norm regularized problem:

$$\min_W \ell(W) + \lambda \|W\|_{\text{tr}} .$$

JS aimed at solving the constrained problem:

$$\min_W \ell(W) \quad \text{s.t.} \quad \|W\|_{\text{tr}} \leq \zeta,$$

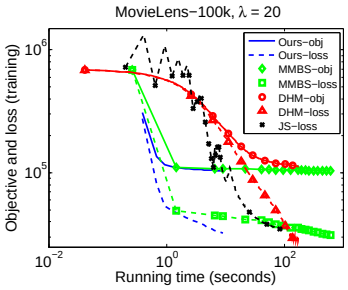
which is hard to directly compare with solvers for the regularized problem. As a workaround, we first chose a λ, and found the optimal solution $W^\star$ for the regularized problem. Then we set $\zeta = \|W^\star\|_{\text{tr}}$ and finally solved the constrained problem by JS. In this case, it is only fair to compare how fast the *loss* $\ell(W)$, rather than the *objective* $\ell(W) + \lambda \|W\|_{\text{tr}}$, is decreased by various solvers. DHM is sensitive to the estimate of the Lipschitz constant of the gradient of $\nabla\ell$, which we manually tuned to a small value such that convergence is still guaranteed. Since the code for MMBS is specialized to the matrix completion problem, it was used only in this comparison. Other solvers such as APG were not included because they are much slower (due to the expensive SVD in each iteration).

4.3.1 Matrix completion

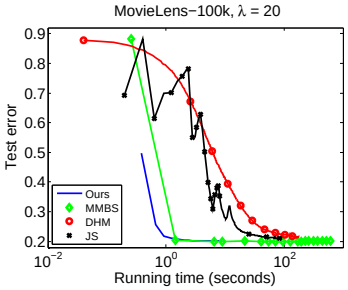
We first compared all methods on a matrix completion problem, using the standard datasets MovieLens100k, MovieLens1M, and MovieLens10M (Jaggi and Sulovsky 2010; Laue 2012; Toh and Yun 2010), which are sized 943×1682 (#user $\times$ #movie), 6040×3706, and 69878×10677 respectively. They contain 10^5, 10^6 and 10^7 movie ratings valued from 1 to 5, and the task is to complete the unobserved entries in the matrix, *i.e.* predict the ratings for some user on unrated movies. The training set was constructed by randomly selecting 50% ratings for each user, and the prediction is made on the rest 50% ratings. We used the square loss. In Figure 4.4 to 4.6, we show how fast various algorithms drive down the training objective, the training loss ℓ, and the normalized mean absolute error (NMAE) on the test set, see, *e.g.* Jaggi and Sulovsky (2010); Toh and Yun (2010). The regularization constant λ is tuned to optimize the test NMAE.

From Figure 4.4(a), 4.5(a), 4.6(a), it is clear that it takes much less amount of CPU time for our method to reduce the objective value (solid line) and the loss ℓ (dashed line). This is due to the effectiveness of our local procedure and the line search in Algorithm 8. Not surprisingly MMBS is the closest to ours in terms of performance because it also adopts local improvement while the other two competitors do not. However, MMBS is still slower because its local search is conducted on a *constrained* manifold. In contrast, our local search surrogate (4.51) is entirely unconstrained and smooth.

[9] http://www.montefiore.ulg.ac.be/ mishra/softwares/traceNorm.html

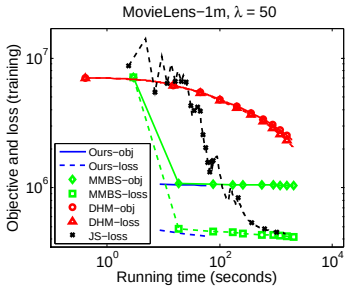


(a) Objective & loss vs time (loglog)

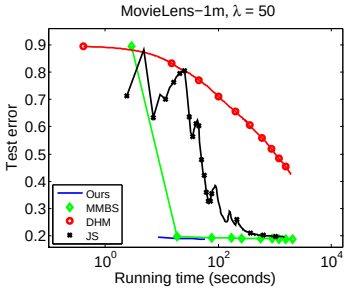


(b) Test NMAE vs time (semilogx)

Figure 4.4: MovieLens100k.

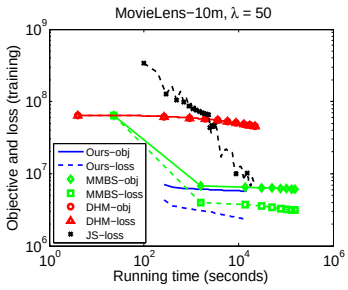


(a) Objective & loss vs time (loglog)

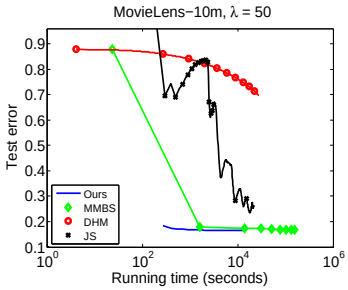


(b) Test NMAE vs time (semilogx)

Figure 4.5: MovieLens1M.



(a) Objective & loss vs time (loglog)



(b) Test NMAE vs time (semilogx)

Figure 4.6: MovieLens10M.

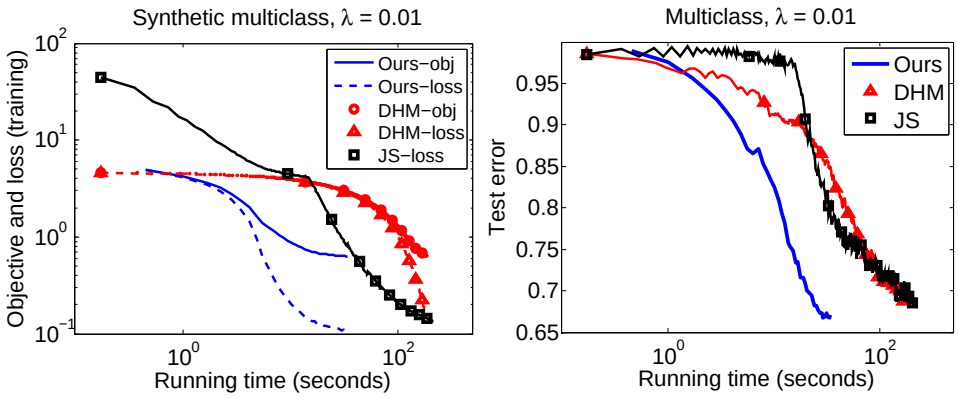
JS, though applied indirectly, is faster than DHM in reducing the loss. We observed that DHM kept running coordinate descent with a constant step size, while its totally corrective update (see *e.g.* (4.37)) was rarely taken. We tried accelerating it by tuning the Lipschitz constant of the gradient of $\nabla\ell$, but this often lead to divergence after a rapid decrease of the objective for the first few iterations.

We also studied the evolution of the NMAE performance on the test data. For this we compared the reconstructed matrix in each iteration against the ground truth. As plotted in Figure 4.4(b), 4.5(b), 4.6(b), our approach achieves comparable (or better) NMAE in much less time than all other methods.

4.3.2 Multi-class and multi-task learning

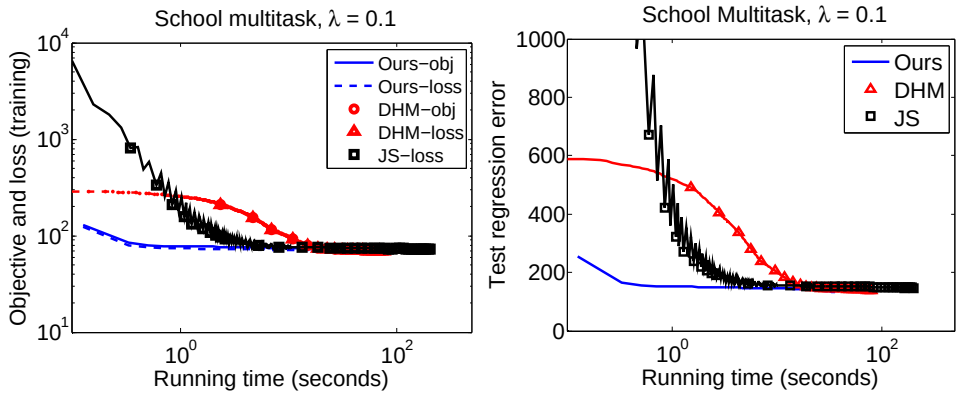
Secondly, we tested on a multi-class classification problem with synthetic dataset. Following Dudik et al. (2012), we generated a dataset of $D = 250$ features and $C = 100$ classes. Each class c has 10 training examples and 10 test examples, drawn independently and identically from a class-specific multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_c, \Sigma_c)$, where the mean $\boldsymbol{\mu}_c \in \mathbb{R}^{250}$ has the last 200 coordinates been 0 and the top 50 coordinates chosen uniformly random from $\{-1, 1\}$, and the (i, j)-th element of the covariance matrix Σ_c is set to $4(0.5)^{|i-j|}$. The goal is to predict the class membership of a given example. We used the logistic loss for a model matrix $W \in \mathbb{R}^{D\times C}$. In particular, for each training example $\mathbf{x}_i$ with label $y_i \in \{1, .., C\}$, we defined an individual loss $\ell_i(W)$ as

$$\ell_i(W) = -\log p(y_i|\mathbf{x}_i; W),$$



(a) Objective & loss vs time (loglog) (b) Test error vs time (semilogx)

Figure 4.7: Multi-class classification on the synthetic dataset.



(a) Objective & loss vs time (loglog) (b) Test error vs time (semilogx)

Figure 4.8: Multi-task learning on the school dataset.

where for any class c,

$$p(c|\mathbf{x}_i; W) = \frac{1}{Z_i} \exp(\mathbf{x}_i^\top W_{:c}), \qquad Z_i = \sum_c \exp(\mathbf{x}_i^\top W_{:c}).$$

Then $\ell(W)$ is defined as the average of $\ell_i(W)$ over the whole training set. We found that $\lambda = 0.01$ yielded the lowest test classification error and the corresponding results are given in Figure 4.7. Clearly, the intermediate models output by our method achieve comparable (or better) training objective and test error in orders of magnitude less time than those generated by DHM and JS.

We also applied the algorithms to a multi-task learning problem on the school dataset of Argyriou et al. (2008). The task is to predict the score of 15362 students from 139 secondary schools based on a number of school-specific and student-specific attributes. Each school is considered as a task for which a predictor is learned. We used the first random split of the training and testing data provided by Argyriou et al. (2008)[10], and set λ so as to achieve the lowest test squared error. Again, as shown

[10] http://ttic.uchicago.edu/~argyriou/code/mtl_feat/school_splits.tar

in Figure 4.8 our approach is much faster than DHM and JS in finding the optimal solution for both the training objective and the test error. As this problem requires a large regularization constant λ, the trace norm regularizer is small, making the loss close to the objective.

4.4 Summary

We have presented a fairly complete overview of the generalized conditional gradient algorithm, which aims at minimizing the sum of a smooth loss and a potentially nonsmooth regularizer. Unlike PG or APG, GCG does not need the proximal map of the regularizer but requires computing the polar instead. In many matrix applications, the latter can be significantly cheaper than the former hence justifying our interest in GCG. We further proposed a GCG variant to handle positively homogeneous regularizers—a common choice in machine learning. Convergence properties of GCG (and the variant) were thoroughly studied. On the application side, we presented a simple relaxation strategy that turns the hard dictionary learning problem into a convex program, which our GCG variant appears to be a natural fit. To further improve the practical performance, we chose to intervene our GCG variant with an effective (fixed-rank) local optimizer, without affecting the convergence property of GCG at all. Finally, we tested our algorithm on two matrix learning problems and validated its practical efficiency.

Chapter 5

Conclusions and Future Directions

We repeat here some of the contributions that we have made in this thesis and point out some possible future directions.

Chapter 1 motivated the composite minimization framework through a sequence of important and familiar examples in machine learning. We then reviewed four popular gradient algorithms all targeted at the composite minimization framework. A common component of these algorithms is the proximal map, which can be computed analytically for simple regularizers such as the ℓ_1 norm. We next demonstrated, through applications that require structured sparse regularizers, that the proximal map may become highly nontrivial thus calls for a detailed study. One interesting direction that is worth further investigation is our different view of the regularized dual averaging algorithm of Xiao (2010). It seems to lead to simplifications and new insights.

Chapter 2 built on some existing works which all suggest that the proximal map of a sum of simple regularizers is merely the composition of the proximal map of each regularizer. We first showed that this observation in general is false, even for projections to closed convex sets. Next we presented a simple sufficient condition on the regularizers so that the suggested prox-decomposition rule is guaranteed to hold. By carefully choosing the right function classes for each regularizer, we aimed at satisfying the sufficient condition by construction, which then allowed us to obtain interesting prox-decompositions. In particular, we proved that a convex function "prox-decomposes" with respect to all gauge functions if and only if it is an increasing function of the Hilbertian norm. Quite unexpectedly, our proof builds on our previous work on characterizing the representer theorem in kernel methods. One thing we are excited about this result is that it may be used to design more sophisticated algorithms that can recover group-wise sparse signals, and to prove deeper convergence results about the proximal gradient algorithm. We also considered the generalization to polyhedral gauge functions that exhibit the cone invariance in their subdifferential and obtained many other prox-decompositions, including some new ones. One interesting observation is that so far our results are either restricted to the "nice" Hilbertian norm-ish function or to polyhedral functions. Whether or not this is a coincidence may be worth some further work. Besides, it might be possible to generalize some of the results in this chapter to nonconvex functions. Finally we mention that in our

related work on characterizing the representer theorem, the characterization in the matrix domain is still incomplete.

Chapter 3 continued our investigation of the proximal map of a sum of simple regularizers. Instead of looking for an exact formula, as we did in Chapter 2, we turned to approximations. In particular, we "pretended" that the nonlinear proximal map is a linear operator, which then makes the computation completely trivial. A bit surprisingly, we proved that this seemingly naive idea not only can be rigorously justified using the proximal average from convex analysis, but also leads to strictly better algorithms than those based on the more familiar "smoothing" idea. A careful inspection of our proposal reveals that we amount to de-smoothing the Moreau envelop—the usual smooth approximation. The benefit is clear: we do not increase the Lipschitz constant. While our actual improvement over the existing approach is secondary, we believe our work is of interest for its clear demonstration of the existence of other effective approximation schemes rather than the familiar smoothing trick, and opens the door for further ideas and possible improvements. In particular, one naturally asks in what sense is a certain approximation scheme optimal? Is there any statistical consequence of our *nonsmooth* approximation? Another conceivable direction is the generalization of our results to non-Hilbertian settings, which would require nontrivial work in extending some convex analytic tools. A very interesting future work, in our opinion, is to abandon the obsession with minimizing a *function* but consider instead finding the zeros of a monotone operator. This brings us some flexibility as in our current work a lot of effort is spent on ensuring ourselves a valid objective function, which is completely off-target. An added bonus is that there exists a vast literature as well as continued advancement on monotone operators that we may draw help from.

The last Chapter 4 considered yet another gradient algorithm, the generalized conditional gradient (GCG). Unlike PG or APG, GCG does not require the proximal map but needs to solve a linear subproblem in each iteration. This linear subproblem, for a gauge regularizer, reduces simply to computing the polar, which, in many matrix applications, can be significantly cheaper than the proximal map. We gave a fairly complete overview of GCG and proposed a variant that handles positively homogeneous regularizers—a common choice in machine learning. We put special focus on establishing various convergence properties of GCG, in particular, we proved its $O(1/t)$ rate of convergence under usual assumptions. Next, we presented a generic convex relaxation strategy to convert the hard dictionary learning problem into a convex program, for which our GCG variant is a convenient candidate solver. To further improve the practical performance, we carefully combined our GCG variant with an effective fixed-rank local search procedure, still retaining the nice convergence properties. Experiments on two matrix learning problems confirmed the effectiveness of our algorithm. As noted in Section 4.2, the polar of the induced regularizer can easily become intractable. However, in many scenarios, it is possible to find a *multiplicatively* approximate solution in reasonable time. Fortunately, we proved that GCG is "robust" enough to accommodate such approximate subroutines, although further work is needed to verify its usefulness. Another open direction

is to extend GCG to nonsmooth losses (other than smoothing), which seems to require significantly new ideas. Results of this type might also be useful in the stochastic or the online setting.

Bibliography

Argyriou, Andreas, Theodoros Evgeniou, and Massimiliano Pontil (2008). "Convex Multi-Task Feature Learning." *Machine Learning*, vol. 73, no. 3, pp. 243–272 (cit. on pp. 84, 92).

Argyriou, Andreas, Rina Foygel, and Nathan Srebro (2012). "Sparse Prediction with the k-Support Norm." In: *Neural Information Processing Systems* (cit. on p. 108).

Argyriou, Andreas, Charles A. Micchelli, and Massimiliano Pontil (2009). "When is There a Representer Theorem? Vector Versus Matrix Regularizers." *Journal of Machine Learning Research*, vol. 10, pp. 2507–2529 (cit. on pp. 39, 40).

Aronszajn, Nachman (1950). "Theory of Reproducing Kernels." *Transactions of the American Mathematical Sociery*, vol. 68 (3), pp. 337–404 (cit. on p. 39).

Arora, Sanjeev, Rong Ge, Ravi Kannan, and Ankur Moitra (2012). "Computing a Nonnegative Matrix Factorization – Provably." In: *ACM Symposium on Theory of Computing* (cit. on p. 84).

Attouch, H (1984). *Variational Convergence for Functions and Operators*. Pitman Adcanced Publishing Program (cit. on pp. 51, 52).

Bach, Francis (2013a). "Convex relaxations of structured matrix factorizations." HAL:00861118 (cit. on p. 79).

— (2013b). "Duality Between Subgradient and Conditional Gradient Methods." arXiv:1211.6302 (cit. on pp. 74, 75).

Bach, Francis, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski (2011). "Optimization with Sparsity-Inducing Penalties." *Foundations and Trends® in Machine Learning*, vol. 4, no. 1, pp. 1–106 (cit. on pp. 10, 47, 48).

— (2012). "Structured Sparsity Through Convex Optimization." *Statistical Science*, vol. 27, no. 4, pp. 450–468 (cit. on pp. 15, 47).

Bach, Francis, Julien Mairal, and Jean Ponce (2008). "Convex Sparse Matrix Factorizations." arXiv:0812.1869v1 (cit. on pp. 84, 87).

Bakin, Sergey (1999). "Adaptive Regression and Model Selection in Data Mining Problems." PhD thesis. Australian National University (cit. on p. 15).

Barzilai, Jonathan and Jonathan M. Borwein (1988). "Two-Point Step Size Gradient Methods." *IMA Journal of Numerical Analysis*, vol. 8, pp. 141–148 (cit. on p. 12).

Bauschke, Heinz H. and Patrick L. Combettes (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer (cit. on pp. 26, 56, 72).

Bauschke, Heinz H., Rafal Goebel, Yves Lucet, and Xianfu Wang (2008). "The Proximal Average: Basic Theory." *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 766–785 (cit. on pp. 32, 47, 53, 54).

Beck, Amir and Marc Teboulle (2009). "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems." *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202 (cit. on pp. 10–12, 47–49).

Bondell, Howard and Brian Reich (2008). "Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR." *Biometrics*, vol. 64, no. 1, pp. 115–123 (cit. on p. 37).

Borwein, Jonathan M. and Adrian S. Lewis (2005). *Convex Analysis and Nonlinear Optimization: Theory and Examples*. 2nd. Springer (cit. on p. 109).

Borwein, Jonathan M. and John D. Vanderwerff (2010). *Convex Functions: Constructions, Characterizations and Counterexamples*. 1st. Cambridge University Press (cit. on p. 69).

Bradley, David M. and J. Andrew Bagnell (2009). "Convex Coding." In: *Conference on Uncertainty in Artificial Intelligence* (cit. on pp. 75, 84).
Bredies, Kristian, Dirk A. Lorenz, and Peter Maass (2009). "A Generalized Conditional Gradient Method and its Connection to an Iterative Shrinkage Method." *Computational Optimization and Applications*, vol. 42, pp. 173–193 (cit. on pp. 66, 69, 70, 81).
Breiman, Leo (1995). "Better Subset Regression Using the Nonnegative Garrote." *Technometrics*, vol. 37, no. 4, pp. 373–384 (cit. on p. 3).
Bühlmann, Peter and Sara van de Geer (2011). *Statistics for High-Dimensional Data*. Springer (cit. on p. 4).
Burer, Samuel and Renato D C Monteiro (2005). "Local Minima and Convergence in Low-Rank Semidefinite Programming." *Mathematical Programming*, vol. 103, no. 3, pp. 427–444 (cit. on p. 88).
Cai, Jian-Feng, Emmanuel J. Candès, and Zuowei Shen (2010). "A Singular Value Thresholding Algorithm for Matrix Completion." *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982 (cit. on p. 19).
Candès, Emmanuel J. and Benjamin Recht (2009). "Exact Matrix Completion via Convex Optimization." *Foundations of Computational Mathematics*, vol. 9, pp. 717–772 (cit. on pp. 18, 19, 110).
Canon, M. D. and C. D. Cullum (1968). "Tight Upper Bound on the Rate of Convergence of Frank-Wolfe Algorithm." *SIAM Journal on Control*, vol. 6, pp. 509–516 (cit. on p. 80).
Chandrasekaran, Venkat, Benjamin Recht, Pablo A. Parrilo, and Alan S.Willsky (2012). "The Convex Geometry of Linear Inverse Problems." *Foundations of Computational Mathematics*, vol. 12, no. 6, pp. 805–849 (cit. on pp. 36, 76, 87).
Chen, Scott Shaobing, David L. Donoho, and Michael A. Saunders (2001). "Atomic Decomposition by Basis Pursuit." *SIAM Review*, vol. 43, no. 1, pp. 129–159. [Originally appeared in SIAM Journal on Scientific Computing, vol. 20, no. 1, pp. 33–61, 1998] (cit. on pp. 4, 5).
Chen, Xi, Qihan Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing (2012). "Smoothing Proximal Gradient Method for General Structured Sparse Regression." *The Annals of Applied Statistics*, vol. 6, no. 2, pp. 719–752 (cit. on pp. 50, 62, 63).
Clarke, Frank H. (1990). *Optimization and Nonsmooth Analysis*. SIAM (cit. on p. 65).
Clarkson, Kenneth L. (2010). "Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm." *ACM Transactions on Algorithms*, vol. 6, no. 4, pp. 1–30 (cit. on pp. 74, 75, 78).
Combettes, Patrick L., Đinh Dũng, and Bằng Công Vũ (2011). "Proximity for Sums of Composite Functions." *Journal of Mathematical Analysis and Applications*, vol. 380, no. 2, pp. 680–688 (cit. on p. 25).
Combettes, Patrick L. and Jean-Christophe Pesquet (2007). "Proximal Thresholding Algorithm for Minimization over Orthonormal Bases." *SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1351–1376 (cit. on pp. 11, 29, 31).
— (2011). "Proximal Splitting Methods in Signal Processing." In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Ed. by Heinz H. Bauschke, Veit Elser, Regina S. Burachik, D. Russell Luke, Patrick L. Combettes, and Henry Wolkowicz. Springer, pp. 185–212 (cit. on pp. 10, 47, 48, 58).
Combettes, Patrick L. and Valérie R. Wajs (2005). "Signal Recovery by Proximal Forward-Backward Splitting." *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200 (cit. on pp. 10, 12, 48).
Condat, Laurent (2013). "A Direct Algorithm for 1D Total Variation Denoising." *IEEE Signal Processing Letters*. to appear (cit. on p. 18).
Cortes, Corinna and Vladimir N. Vapnik (1995). "Support-Vector Networks." *Machine Learning*, vol. 20, no. 3, pp. 273–297 (cit. on p. 2).
Daubechies, Ingrid, Michel Defrise, and Christine De Mol (2004). "An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint." *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457 (cit. on p. 11).
Dem'yanov, V. F. and A. M. Rubinov (1967). "The Minimization of a Smooth Convex Functional on a Convex Set." *SIAM Journal on Control*, vol. 5, pp. 280–294. [English translation of paper

in *Vestnik Leningradskogo Universitera, Seriya Matematiki, Mekhaniki i Astronomii* vol. 19, pp. 7–17, 1964] (cit. on pp. 66, 69).

Devroye, Luc, László Györfi, and Gábor Lugosi (1996). *A Probabilistic Theory of Pattern Recognition*. Springer (cit. on p. 1).

Dinuzzo, Francesco and Bernhard Schölkopf (2012). "The Representer Theorem for Hilbert Apaces: A Necessary and Sufficient Condition." In: *Advances in Neural Information Processing Systems* (cit. on pp. 23, 39, 40, 44).

Donoho, David L. and Xiaoming Huo (2001). "Uncertainty Principles and Ideal Atomic Decomposition." *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862 (cit. on p. 5).

Duchi, John C., Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari (2010). "Composite Objective Mirror Descent." In: *Conference on Learning Theory* (cit. on p. 47).

Duchi, John C. and Yoram Singer (2009). "Efficient Online and Batch Learning Using Forward Backward Splitting." *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934 (cit. on pp. 13, 38).

Dudik, Miroslav, Zaid Harchaoui, and Jerome Malick (2012). "Lifted Coordinate Descent for Learning with Trace-norm Regularizations." In: *Conference on Artificial Intelligence and Statistics* (cit. on pp. 80, 90, 91).

Efron, Bradley, Trevor Hastie, Iain Johnstone, and Robert Tibshirani (2004). "Least Angle Regression." *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499 (cit. on pp. 16, 17).

Ekeland, Ivar and Roger Témam (1999). *Convex Analysis and Variational Problems*. SIAM (cit. on pp. 25, 27).

Fazel, Maryam, Haitham Hindi, and Stephen P. Boyd (2001). "A Rank Minimization Heuristic with Application to Minimum Order System Approximation." In: *American Control Conference*, pp. 4734–4739 (cit. on p. 107).

Figueiredo, Mário A. T. and Robert D. Nowak (2003). "An EM Algorithm for Wavelet-Based Image Restoration." *IEEE Transactions on Signal Processing*, vol. 12, no. 8, pp. 906–916 (cit. on p. 11).

Foucart, Simon and Holger Rauhut (2013). *A Mathematical Introduction to Compressive Sensing*. Springer (cit. on p. 5).

Frank, Marguerite and Philip Wolfe (1956). "An Algorithm for Quadratic Programming." *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110 (cit. on pp. 20, 66, 69, 74).

Freund, Robert M. and Paul Grigas (2013). "New Analysis and Results for the Conditional Gradient Method." preprint (cit. on p. 74).

Freund, Yoav and Robert E. Schapire (1997). "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting." *Journal of Computer and System Sciences*, vol. 55, pp. 119–139 (cit. on p. 82).

Friedman, Jerome, Trevor Hastie, Holger Höfling, and Robert Tibshirani (2007). "Pathwise Coordinate Optimization." *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332 (cit. on pp. 17, 18, 20, 23, 37).

Fukushima, Masao and Hisashi Mine (1981). "A Generalized Proximal Point Algorithm for Certain Non-Convex Minimization Problems." *International Journal of Systems Science*, vol. 12, no. 8, pp. 989–1000 (cit. on pp. 9, 52).

Goldstein, A. A. (1964). "Convex Programming in Hilbert Space." *Bulletin of the American Mathematical Society*, vol. 70, no. 5, pp. 709–710 (cit. on pp. 10, 12).

Grove, Adam J. and Dale Schuurmans (1998). "Boosting in the Limit: Maximizing the Margin of Learned Ensembles." In: *AAAI Conference on Artificial Intelligence* (cit. on p. 83).

Güler, Osman (1991). "On the Convergence of the Proximal Point Algorithm for Convex Minimization." *SIAM Journal on Control and Optimization*, vol. 29, no. 2, pp. 403–419 (cit. on p. 11).

Ham, Jihun, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf (2004). "A Kernel View of the Dimensionality Reduction of Manifolds." In: *International Conference on Machine Learning*, pp. 369–376 (cit. on p. 110).

Hardy, G. H., J. E. Littlewood, and G. Pólya (1952). *Inequalities*. 2nd. Cambridge University Press (cit. on p. 72).

Hazan, Elad (2008). "Sparse Approximate Solutions to Semidefinite Programs." In: *Latin American Conference on Theoretical Informatics* (cit. on pp. 75, 78).

Hebiri, Mohamed and Sara van de Geer (2011). "The Smoothed-Lasso and other $\ell_1 + \ell_2$-Penalized Methods." *Electronic Journal of Statistics*, vol. 5, pp. 1184–1226 (cit. on p. 17).
Hoefling, Holger (2010). "A Path Algorithm for the Fused Lasso Signal Approximator." *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 984–1006 (cit. on pp. 18, 47).
Huber, Peter J. (1964). "Robust Estimation of a Location Parameter." *Annals of Mathematical Statistics*, vol. 35, pp. 73–101 (cit. on pp. 10, 107).
Jaggi, Martin (2013). "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization." In: *International Conference on Machine Learning* (cit. on pp. 74, 75, 78).
Jaggi, Martin and Marek Sulovsky (2010). "A Simple Algorithm for Nuclear Norm Regularized Problems." In: *International Conference on Machine Learning* (cit. on pp. 75, 78, 90).
Jenatton, Rodolphe, Julien Mairal, Guillaume Obozinski, and Francis Bach (2011). "Proximal Methods for Hierarchical Sparse Coding." *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334 (cit. on pp. 17, 20, 23, 35, 38).
Jojic, Vladimir, Suchi Saria, and Daphne Koller (2011). "Convex Envelopes of Complexity Controlling Penalties: the Case Against Premature Evelopment." In: *AISTAT* (cit. on p. 107).
Kim, Seung-Jean, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky (2009). "ℓ_1 Trend Filtering." *SIAM Review*, vol. 51, no. 2, pp. 339–360 (cit. on pp. 17, 18).
Kim, Seyoung and Eric P. Xing (2009). "Statistical Estimation of Correlated Genome Associations to a Quantitative Trait Network." *PLoS Genetics*, vol. 5, no. 8, pp. 1–18 (cit. on pp. 18, 47, 50).
Kimeldorf, George and Grace Wahba (1971). "Some Results on Tchebycheffian Spline Functions." *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95 (cit. on p. 39).
Korte, Bernhard and Jens Vygen (2012). *Combinatorial Optimization: Theory and Algorithms*. 5th. Springer (cit. on p. 36).
Laue, Soren (2012). "A Hybrid Algorithm for Convex Semidefinite Optimization." In: *International Conference on Machine Learning* (cit. on pp. 89, 90).
Lee, Daniel D and H Sebastian Seung (1999). "Learning the Parts of Objects by Non-Negative Matrix Factorization." *Nature*, vol. 401, pp. 788–791 (cit. on p. 84).
Levitin, E. S. and B. T. Polyak (1966). "Constrained Minimization Problems." *USSR Computational Mathematics and Mathematical Physics*, vol. 6, no. 5, pp. 1–50. [English translation of paper in *Zh. Vȳchisl. Mat. mat. Fiz.* vol. 6, no. 5, pp. 787–823, 1965] (cit. on pp. 66, 69, 71, 74).
Ma, Shiqian, Donald Goldfarb, and Lifeng Chen (2011). "Fixed Point and Bregman Iterative Methods for Matrix Rank Minimization." *Mathematical Programming, Series A*, vol. 128, pp. 321–353 (cit. on p. 19).
Mallat, Stéphane (2009). *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd. Elsevier (cit. on pp. 5, 83).
Martinet, B. (1970). "Régularisation d'Inéquations Variationnelles par Approximations Successives." *Revue Française d'Informatique et de Recherche Opérationnelle, Série Rouge*, vol. 4, no. 3, pp. 154–158 (cit. on pp. 9, 51, 52).
Martins, André F. T., Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo (2011). "Online Learning of Structured Predictors with Multiple Kernels." In: *Conference on Artificial Intelligence and Statistics* (cit. on p. 25).
Mason, Llew, Jonathan Baxter, Peter L. Bartlett, and Marcus Frean (2000). "Functional Gradient Techniques for Combining Hypotheses." In: *Advances in Large Margin Classifiers*, pp. 221–246 (cit. on p. 83).
Meyer, Gerard (1974). "Accelerated Frank–Wolfe Algorithms." *SIAM Journal on Control*, vol. 12, pp. 655–655 (cit. on p. 80).
Miller, Alan (2002). *Subset Selection in Regression*. 2nd. Chapman & Hall/CRC (cit. on p. 2).
Mine, Hisashi and Masao Fukushima (1981). "A Minimization Method for the Sum of a Convex Function and a Continuously Differentiable Function." *Journal of Optimization Theory and Applications*, vol. 33, no. 1, pp. 9–23 (cit. on pp. 66, 69, 70).
Mishra, Bamdev, Gilles Meyer, Francis Bach, and Rodolphe Sepulchre (2013). "Low-rank Optimization with Trace Norm Penalty." *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2124–2149 (cit. on pp. 89, 90).

Moreau, Jean J. (1965). "Proximité et Dualtité dans un Espace Hilbertien." *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299 (cit. on pp. 10, 12, 23–25, 28, 32, 51, 53).
Natarajan, B. K. (1995). "Sparse Approximate Solutions to Linear Systems." *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234 (cit. on pp. 3, 106).
Nesterov, Yurii (1983). "A Method for Solving a Convex Programming Problem with Convergence Rate $O(1/k^2)$." *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376 (cit. on p. 11).
— (2003). *Introductory Lectures on Convex Optimization: A Basic Course*. Springer (cit. on pp. 2, 8, 9).
— (2005). "Smooth Minimization of Non-Smooth Functions." *Mathematical Programming, Series A*, vol. 103, pp. 127–152 (cit. on pp. 47–49, 57).
— (2009). "Primal-Dual Subgradient Methods for Convex Problems." *Mathematical Programming, Series B*, vol. 120, pp. 221–259 (cit. on p. 13).
— (2013). "Gradient Methods for Minimizing Composite Functions." *Mathematical Programming, Series B*, vol. 140, pp. 125–161 (cit. on pp. 10, 12, 47, 48).
Nesterov, Yurii and Arkadi Nemiroviskii (1994). *Interior-point Polynomial Methods in Convex Programming*. SIAM (cit. on p. 6).
Olshausen, Bruno A. and David J. Field (1996). "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images." *Nature*, vol. 381, pp. 607–609 (cit. on pp. 5, 83).
Owen, Art B. (2007). "A Robust Hybrid of Lasso and Ridge Regression." In: *Prediction and Discovery*. AMS, pp. 59–72 (cit. on p. 34).
Parikh, Neal and Stephen Boyd (2013). "Proximal Algorithms." *Foundations and Trends®in Optimization*, vol. 1, no. 3, pp. 123–231 (cit. on pp. 10, 47, 48).
Pong, Ting Kei, Paul Tseng, Shuiwang Ji, and Jieping Ye (2010). "Trace Norm Regularization: Reformulations, Algorithms, and Multi-task Learning." *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3465–3489 (cit. on p. 19).
Portnoy, Stephen and Roger Koenker (1997). "The Gaussian Hare and the Laplacian Tortoise: Computability of Squared-Error versus Absolute-Error Estimators." *Statistical Science*, vol. 12, no. 4, pp. 279–300 (cit. on p. 6).
Recht, Ben, Christopher Re, Joel Tropp, and Victor Bittorf (2012). "Factoring Nonnegative Matrices with Linear Programs." In: *Advances in Neural Information Processing Systems 25* (cit. on p. 84).
Rockafellar, Ralph Tyrell and Roger J-B Wets (1998). *Variational Analysis*. Springer (cit. on pp. 50–52, 54).
Rockafellar, Ralph Tyrrell (1976). "Monotone Operators and The Proximal Point Algorithm." *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898 (cit. on pp. 9, 51, 52).
Roweis, Sam T. and Lawrence K. Saul (2000). "Nonlinear Dimensionality Reduction by Locally Linear Embedding." *Science*, vol. 290, pp. 2323–2326 (cit. on p. 110).
Rudin, Leonid I., Stanley Osher, and Emad Fatemi (1992). "Nonlinear Total Variation Based Noise Removal Algorithms." *Physica D*, vol. 60, pp. 259–268 (cit. on p. 17).
Rudin, Walter (1976). *Principles of mathematical analysis*. 3rd. McGraw-Hill (cit. on p. 46).
Scholköpf, Bernhard and Alexander J. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press (cit. on p. 39).
Shalev-Shwartz, Shai and Yoram Singer (2010). "On the Equivalence of Weak Learnability and Linear Separability: New Relaxations and Efficient Boosting Algorithms." *Machine Learning*, vol. 80, pp. 141–163 (cit. on p. 83).
Shalev-Shwartz, Shai, Nathan Srebro, and Tong Zhang (2010). "Trading Accuracy for Sparsity in Optimization Problems with Sparsity Constraints." *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2807–2832 (cit. on pp. 75, 78, 80).
Shamir, Ohad and Tong Zhang (2013). "Stochastic Gradient Descent for Non-Smooth Optimization: Convergence Results and Optimal Averaging Schemes." In: *International Conference on Machine Learning* (cit. on p. 74).
Shaw, Blake and Tony Jebara (2007). "Minimum Volume Embedding." In: *Conference on Artificial Intelligence and Statistics*, pp. 460–467 (cit. on p. 111).

Shor, Naum Z. (1985). *Minimization Methods for Non-Differentiable Functions*. Springer (cit. on pp. 6, 47, 48).
Sra, Suvrit, Sebastian Nowozin, and Stephen J. Wright (2012). *Optimization for Machine Learning*. MIT Press (cit. on p. 6).
Srebro, Nathan, Jason D. M. Rennie, and Tommi S. Jaakkola (2005). "Maximum-Margin Matrix Factorization." In: *Advances in Neural Information Processing Systems* (cit. on p. 88).
Starck, Jean-Luc, David L. Donoho, and Emmanuel J. Candès (2003). "Astronomical Image Representation by the Curvelet Transform." *Astronomy & Astrophysics*, vol. 398, pp. 785–800 (cit. on p. 11).
Steinwart, Ingo (2007). "How to Compare Different Loss Functions and Their Risks." *Constructive Approximation*, vol. 26, pp. 225–287 (cit. on p. 2).
Steinwart, Ingo and Andreas Christmann (2008). *Support Vector Machines*. Springer (cit. on p. 2).
Stigler, Stephen M. (1984). "Boscovich, Simpson and A 1760 Manuscript Note on Fitting a Linear Relation." *Biometrika*, vol. 71, no. 3, pp. 615–620 (cit. on p. 6).
Temlyakov, Vladimir (2011). *Greedy Approximation*. Cambridge University Press (cit. on p. 83).
Tenenbaum, Joshua B., Vin de Silva, and John C. Langford (2000). "A Global Geometric Framework for Nonlinear Dimensionality Reduction." *Science*, vol. 290, pp. 2319–2323 (cit. on p. 110).
Tewari, Ambuj, Pradeep Ravikumar, and Inderjit S. Dhillon (2011). "Greedy Algorithms for Structurally Constrained High Dimensional Problems." In: *Advances in Neural Information Processing Systems* (cit. on pp. 75, 78, 87).
Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288 (cit. on pp. 3, 4).
— (2011). "Regression Shrinkage and Selection via the Lasso: A Retrospective." *Journal of the Royal Statistical Society, Series B*, vol. 73, pp. 273–282 (cit. on p. 6).
Tibshirani, Robert, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight (2005). "Sparsity and Smoothness via the Fused Lasso." *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 91–108 (cit. on pp. 17, 50).
Toh, Kim-Chuan and Sangwoon Yun (2010). "An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Least Squares Problems." *Pacific Journal of Optimization*, vol. 6, pp. 615–640 (cit. on pp. 19, 90).
Trèves, François (1967). *Topological Vector Spaces, Distributions and Kernels*. Dover Reprint (cit. on p. 65).
Tseng, Paul (2008). "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization" (cit. on pp. 10–12).
— (2010). "Approximation Accuracy, Gradient Methods, and Error Bound for Structured Convex Optimization." *Mathematical Programming, Series B*, vol. 125, pp. 263–295 (cit. on pp. 10, 11).
Vavasis, Stephen A (2010). "On the Complexity of Nonnegative Matrix Factorization." *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1364–1377 (cit. on p. 84).
Villa, Silvia, Saverio Salzo, Luca Baldassarre, and Alessandro Verri (2013). "Accelerated and Inexact Forward-Backward Algorithms." *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1607–1633 (cit. on pp. 47, 48).
von Neumann, John (1937). "Some Matrix-Inequalities and Metrization of Matric-Space." *Tomsk. Univ. Rev*, vol. 1, pp. 286–300 (cit. on pp. 19, 109).
Warmuth, Manfred K., Karen A. Glocer, and S. V. N. Vishwanathan (2008). "Entropy Regularized LPBoost." In: *Conference on Algorithmic Learning Theory* (cit. on p. 83).
Weinberger, Kilian Q. and Lawrence K. Saul (2006). "Unsupervised Learning of Image Manifolds by Semidefinite Programming." *International Journal of Computer Vision*, vol. 70, no. 1, pp. 77–90 (cit. on p. 110).
White, Martha, Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans (2012). "Convex Multi-view Subspace Learning." In: *Advances in Neural Information Processing Systems* (cit. on pp. 64, 86).
Wright, Stephen J., Mário A. T. Figueiredo, and Robert D. Nowak (2009). "Sparse Reconstruction by Separable Approximation." *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493 (cit. on p. 12).

Xiao, Lin (2010). "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization." *Journal of Machine Learning Research*, vol. 11, pp. 2543–2596 (cit. on pp. 13, 14, 47, 94).

Yu, Yaoliang (2013a). "Better Approximation and Faster Algorithm Using the Proximal Average." In: *Advances in Neural Information Processing Systems* (cit. on pp. 21, 46).

— (2013b). "On Decomposing the Proximal Map." In: *Advances in Neural Information Processing Systems* (cit. on pp. 21, 22).

Yu, Yaoliang, Hao Cheng, Dale Schuurmans, and Csaba Szepesvári (2013). "Characterizing the Representer Theorem." In: *International Conference on Machine Learning* (cit. on pp. 21–23, 44).

Yu, Yaoliang, James Neufeld, Ryan Kiros, Xinhua Zhang, and Dale Schuurmans (2012). "Regularizers versus Losses for Nonlinear Dimensionality Reduction." In: *International Conference on Machine Learning* (cit. on p. 112).

Yu, Yaoliang and Dale Schuurmans (2011). "Rank/Norm Regularization with Closed-Form Solutions: Application to Subspace Clustering." In: *Conference on Uncertainty in Artificial Intelligence* (cit. on pp. 19, 38, 84, 109).

Yuan, Ming and Yi Lin (2006). "Model Selection and Estimation in Regression with Grouped Variables." *Journal of the Royal Statistical Society, Series B*, vol. 68, pp. 49–67 (cit. on pp. 15, 16).

Yuan, Xiaotong and Shuicheng Yan (2013). "Forward Basis Selection for Pursuing Sparse Representations Over a Dictionary." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. to appear (cit. on pp. 75, 78, 80).

Zălinescu, C. (2002). *Convex Analysis in General Vector Spaces*. World Scientific (cit. on pp. 7, 9, 14, 52, 75, 104, 108).

Zhang, Tong (2003). "Sequential Greedy Approximation for Certain Convex Optimization Problems." *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 682–691 (cit. on p. 83).

Zhang, Xinhua, Yaoliang Yu, and Dale Schuurmans (2012). "Accelerated Training for Matrix-Norm Regularization: A Boosting Approach." In: *Advances in Neural Information Processing Systems* (cit. on pp. 21, 64, 84, 86).

— (2013). "Polar Operators for Structured Sparse Estimation." In: *Advances in Neural Information Processing Systems* (cit. on pp. 18, 37).

Zhang, Xinhua, Yaoliang Yu, Martha White, Ruitong Huang, and Dale Schuurmans (2011). "Convex Sparse Coding, Subspace Learning, and Semi-Supervised Extensions." In: *AAAI Conference on Artificial Intelligence* (cit. on pp. 64, 84, 86).

Zhao, Peng, Guilherme Rocha, and Bin Yu (2009). "The Composite Absolute Penalties Family for Grouped and Hierarchical Variable Selection." *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497 (cit. on pp. 16, 17, 47, 50).

Zhong, Leon Wenliang and James T. Kwok (2011). "Efficient Sparse Modeling with Automatic Feature Grouping." In: *International Conference on Machine Learning* (cit. on p. 37).

Zhou, Jiayu, Jun Liu, Vaibhav A. Narayan, and Jieping Ye (2012). "Modeling Disease Progression via Fused Sparse Group Lasso." In: *Conference on Knowledge Discovery and Data Mining* (cit. on pp. 23, 27).

Zou, Hui (2006). "The Adaptive Lasso and Its Oracle Properties." *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429 (cit. on p. 109).

Zou, Hui and Trevor Hastie (2005). "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society B*, vol. 67, pp. 301–320 (cit. on pp. 33, 107).

Appendix A

Constructing Convex Regularizers

We turn our attention in this appendix to constructing convex relaxations for some highly nonconvex, combinatorial regularizers. The main idea is to employ the biconjugate function from convex analysis, which we detail in the following. We provide three examples for illustration.

As before let our domain $\mathcal{H}$ be a (real) Hilbert space and consider extended real-valued functions $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. The (closed) convex hull of an arbitrary function f is defined as the *greatest* (closed)[1] convex function that minorizes f. Since taking pointwise supremum preserves convexity, the (closed) convex hull indeed exists: just collect all (closed) convex functions that minorize f and take their pointwise supremum. We use $\operatorname{conv} f$ and $\overline{\operatorname{conv}} f$ to denote the convex hull and closed convex hull of the function f, respectively. Similarly we use $\operatorname{conv} C$ and $\overline{\operatorname{conv}} C$ to denote the convex hull and closed convex hull of the point set C, respectively. Conveniently, study of functions can be reduced to study of point sets through the epigraph construction $\operatorname{epi} f := \{(\mathbf{w}, t) \in \mathcal{H} \times \mathbb{R} : f(\mathbf{w}) \leq t\}$. It is easy to prove that f is (closed) convex iff $\operatorname{epi} f$ is (closed) convex.

While there are multiple equivalent characterizations of the closed convex hull, perhaps the most friendly one is through the Fenchel conjugate, defined as:

$$f^*(\mathbf{w}^*) = \sup_{\mathbf{w}} \langle \mathbf{w}, \mathbf{w}^* \rangle - f(\mathbf{w}), \tag{A.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathcal{H}$. Note that for any function f, not necessarily convex, its Fenchel conjugate is always closed and convex since by its definition it is the pointwise supremum of affine, *bona fide* closed and convex, functions. A standard duality argument reveals that the closed convex hull is exactly the Fenchel biconjugate (Zălinescu 2002, Theorem 2.3.4):

$$\overline{\operatorname{conv}} f = f^{**}. \tag{A.2}$$

Apparently, if f is closed and convex, then $f = \overline{\operatorname{conv}} f = f^{**}$.

Since the (closed) convex hull is the *uniformly* tightest convex underestimate, it makes sense to replace a "hard" function with its (closed) convex hull as the latter is usually much easier to

[1] Recall that a function f is closed iff its sublevel sets $\{\mathbf{w} \in \mathcal{H} : f(\mathbf{w}) \leq \alpha\}$ are closed for all $\alpha \in \mathbb{R}$.

minimize. The machine learning community has witnessed increasing interest and surprising effectiveness of this seemingly simple trick. We will present three case studies to confirm this point, but before that let us first review some basic properties about the closed convex hull.

A.1 Some Basic Results

We record here a few basic results about the closed convex hull, some of which will become handy in later calculations. The proofs are straightforward hence omitted.

Lemma A.1. *If $f \geq g$ then $f^* \leq g^*$, hence $\overline{\operatorname{conv}} f \geq \overline{\operatorname{conv}} g$ and $\operatorname{conv} f \geq \operatorname{conv} g$.*

Recall that $\operatorname{dom} f$ denotes the *effective* domain of the function f, i.e., the set $\{\mathbf{w} \in \mathcal{H} : f(\mathbf{w}) < \infty\}$.

Lemma A.2. $\operatorname{conv} f \geq \overline{\operatorname{conv}} f$ *and* $\operatorname{conv}(\operatorname{dom} f) = \operatorname{dom}(\operatorname{conv} f) \subseteq \operatorname{dom}(\overline{\operatorname{conv}} f) \subseteq \overline{\operatorname{conv}}(\operatorname{dom} f)$.

The last two inclusions may be strict. In a finite dimensional setting, if $\operatorname{dom} f$ is compact, then $\operatorname{conv}(\operatorname{dom} f) = \overline{\operatorname{conv}}(\operatorname{dom} f)$, hence we will have only equalities in the above lemma.

Lemma A.3. *Let $\mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_k$, $f_i : \mathcal{H}_i \to \mathbb{R} \cup \{\infty\}$ and $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. If $f := \sum_{i=1}^{k} f_i$, then $f^* = \sum_{i=1}^{k} f_i^*$ hence $\overline{\operatorname{conv}} f = \sum_{i=1}^{k} \overline{\operatorname{conv}} f_i$.*

The next three lemmas are immediate consequences of Fenchel duality.

Lemma A.4. $\forall \lambda > 0$, $\overline{\operatorname{conv}}(\lambda f) = \lambda \, \overline{\operatorname{conv}} f$ *and* $\operatorname{conv}(\lambda f) = \lambda \, \operatorname{conv} f$.

Lemma A.5. $\overline{\operatorname{conv}}(f + \langle \cdot, \mathbf{a} \rangle + \alpha) = \overline{\operatorname{conv}} f + \langle \cdot, \mathbf{a} \rangle + \alpha$ *and* $\operatorname{conv}(f + \langle \cdot, \mathbf{a} \rangle + \alpha) = \operatorname{conv} f + \langle \cdot, \mathbf{a} \rangle + \alpha$.

Lemma A.5 is no longer true even when we replace the affine function with some *convex* function. The reason, intuitively, is because the convex part might transfer some "extra convexity" into the nonconvex part. An explicit example can be found in the next section.

Lemma A.6. $\overline{\operatorname{conv}}(f(A \cdot + \mathbf{b})) = (\overline{\operatorname{conv}} f)(A \cdot + \mathbf{b})$ *and* $\operatorname{conv}(f(A \cdot + \mathbf{b})) = (\operatorname{conv} f)(A \cdot + \mathbf{b})$, *where $A : \mathcal{H} \to \mathcal{H}$ is an invertible linear map.*

For the next two lemmas only, we allow our functions to take value $-\infty$, or one may simply assume the infimum is finite.

Lemma A.7. *If $\sup_{\mathbf{w} \in \mathcal{H}} f(\mathbf{w}) < \infty$ then $\operatorname{conv} f = \overline{\operatorname{conv}} f \equiv \inf_{\mathbf{w} \in \mathcal{H}} f(\mathbf{w})$.*

Lemma A.7 is important in the following sense: When our "hard" function f is bounded from above, it is meaningless to naively replace it with its (closed) convex hull. We must somehow first make f unbounded, which is usually done by adding some "reasonable" unbounded function.

Our last result explains why closed convex hulls are so useful in nonconvex optimization.

Lemma A.8. *Consider an arbitrary function $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$, we have*

$$OPT := \inf_{\mathbf{w}} f(\mathbf{w}) = \inf_{\mathbf{w}} (\overline{\text{conv}} f)(\mathbf{w}) \tag{A.3}$$

and

$$\overline{\text{conv}}\{\mathbf{w} : f(\mathbf{w}) = OPT\} \subseteq \{\mathbf{w} : (\overline{\text{conv}} f)(\mathbf{w}) = OPT\}. \tag{A.4}$$

The inclusion may be strict, and an example can be easily constructed with the help of Lemma A.7. We emphasize that Lemma A.8 does *not* free us from minimizing nonconvex functions since usually it is not at all easier to construct the closed convex hull.

Equipped with these technical results, we are now ready to present some examples.

A.2 Example 1: Sparsity

The first function we consider is the cardinality function:

$$\|\mathbf{w}\|_0 := \sum_{i=1}^{m} \mathbb{1}\{w_i \neq 0\}, \tag{A.5}$$

where we use $\mathbb{1}\{w_i \neq 0\}$ to denote the $\{0, 1\}$-valued indicator function. The cardinality function is a perfect regularizer if a sparse solution is desired. Recent years have witnessed the flourish of sparsity-targeted methods, most notably the LASSO in Example 1.4 and the basis pursuit in Example 1.5 (both discussed in Chapter 1). Unfortunately, minimizing the cardinality function, even subject to linear constraints, is NP-Hard (Natarajan 1995).

A very natural idea is to replace the "hard" cardinality function with its closed convex hull, however, Lemma A.7 tells us that the latter in this case is trivially the constant zero function. To get a meaningful convex hull, we consider adding some "reasonable" unbounded (from above) function to $\|\cdot\|_0$.

A.2.1 ℓ_p-Norm Regularization

Recall that the ℓ_p norm is defined as $\|\mathbf{w}\|_p := (\sum_i |w_i|^p)^{1/p}$. To avoid a trivial convex hull, we consider adding an ℓ_p norm to $\|\cdot\|_0$ (for $1 \leq p < \infty$):

$$\|\cdot\|_{p+0} := \|\cdot\|_0 + \frac{\lambda}{p}\|\cdot\|_p^p. \tag{A.6}$$

The reason to take the p-th power of the ℓ_p norm is merely for computational convenience. From an optimization point of view, taking the p-th power, or not, is equivalent up to an appropriate change of the constant λ.

Observe that by Lemma A.3, we only need to deal with the univariate case. Straightforward calculation verifies that for $1 < p < \infty$:

$$\overline{\text{conv}}(\|\cdot\|_{p+0})(\mathbf{w}) = (\|\cdot\|_{p+0})^{**}(\mathbf{w}) = \sum_{i=1}^{m} \max\left\{\frac{\lambda}{p}|w_i|^p + 1,\ \lambda^{1/p} q^{1/q} |w_i|\right\}, \tag{A.7}$$

where as usual q is the conjugate exponent of p, *i.e.*, $1/p + 1/q = 1$. While for $p = 1$, we have

$$\overline{\text{conv}}(\|\cdot\|_{1+0})(\mathbf{w}) = (\|\cdot\|_{1+0})^{**}(\mathbf{w}) = \lambda\|\mathbf{w}\|_1. \tag{A.8}$$

This partly explains why the ℓ_1 norm is a good convex surrogate for the cardinality function (A.5). And lastly for

$$\|\mathbf{w}\|_{\infty+0} := \|\mathbf{w}\|_0 + \lambda\|\mathbf{w}\|_\infty, \tag{A.9}$$

we have

$$\overline{\text{conv}}(\|\cdot\|_{\infty+0})(\mathbf{w}) = (\|\cdot\|_{\infty+0})^{**}(\mathbf{w}) = \lambda\|\mathbf{w}\|_\infty. \tag{A.10}$$

Observe that the convex hull in (A.7) is very similar to Huber's loss in robust statistics (Huber 1964). When $p = 2$, similar derivation as here has appeared in Jojic et al. (2011), who also argued that (A.7) is tighter/better than the elastic net regularizer of Zou and Hastie (2005), that is, $\|\cdot\|_1 + \frac{\lambda}{2}\|\cdot\|_2^2$.

Apparently, $\overline{\text{conv}}(\|\cdot\|_0) + \frac{\lambda}{p}\|\cdot\|_p^p = \frac{\lambda}{p}\|\cdot\|_p^p \neq \overline{\text{conv}}(\|\cdot\|_0 + \frac{\lambda}{p}\|\cdot\|_p^p)$. This is the example we mentioned after Lemma A.5.

A.2.2 Truncation

Next, we consider truncating the cardinality function:

$$\|\mathbf{w}\|_{\infty\wedge 0} := \|\mathbf{w}\|_0 + \iota_{\{\|\cdot\|_\infty \leq 1\}}(\mathbf{w}), \tag{A.11}$$

which is equivalent as adding the $\{0, \infty\}$-valued indicator function $\iota_C(\mathbf{w})$, *i.e.*, 0 if $\mathbf{w} \in C$ and ∞ otherwise. Historically, this is how the trace norm regularizer (*i.e.*, the ℓ_1-norm in the matrix sense) was first derived in Fazel et al. (2001). This truncation idea has some advantage over the addition of the ℓ_p norm, as we will see at the end of this subsection.

Thanks to Lemma A.3 which allows us to reduce to the univariate case, it is easy to verify

$$\overline{\text{conv}}(\|\cdot\|_{\infty\wedge 0})(\mathbf{w}) = (\|\cdot\|_{\infty\wedge 0})^{**}(\mathbf{w}) = \|\mathbf{w}\|_1 + \iota_{\{\|\cdot\|_\infty \leq 1\}}(\mathbf{w}), \tag{A.12}$$

i.e. the ℓ_1 norm restricted to the ℓ_∞ norm unit ball. Naturally, one wonders what would happen if we truncate the cardinality function differently, for instance, instead of restricting to the ℓ_∞ ball, how about the ℓ_p ball? The result turns out to be somewhat complicated.

Before addressing the general case, let us compute the convex hull for a rather peculiar truncation:

$$\|\cdot\|_{1\wedge 0} := \|\mathbf{w}\|_0 + \iota_{\{0, \pm\mathbf{e}_1, \ldots, \pm\mathbf{e}_m\}}(\mathbf{w}), \tag{A.13}$$

where $\{\mathbf{e}_i\}$ form the canonical basis for $\mathbb{R}^m$. From Lemma A.2 we know that the closed convex hull is defined on $\text{conv}(\{0, \pm\mathbf{e}_1, \ldots, \pm\mathbf{e}_m\}) = \{\|\mathbf{x}\|_1 \leq 1\}$. Easy calculation shows that

$$\overline{\text{conv}}(\|\cdot\|_{1\wedge 0})(\mathbf{w}) = \|\mathbf{w}\|_1 + \iota_{\{\|\cdot\|_1 \leq 1\}}(\mathbf{w}), \tag{A.14}$$

the restriction of the ℓ_1 norm to its own unit ball.

More generally, consider

$$\|\cdot\|_{C\wedge 0} := \|\mathbf{w}\|_0 + \iota_C(\mathbf{w}),$$

where C is any closed set satisfying $\{0, \pm\mathbf{e}_1, \ldots, \pm\mathbf{e}_m\} \subseteq C \subseteq \{\|\cdot\|_\infty \leq 1\}$. Lemma A.2 implies that the effective domain of $\overline{\operatorname{conv}}(\|\cdot\|_{\infty\wedge 0})$ is $\operatorname{conv} C$, and by Lemma A.1, we have

$$\overline{\operatorname{conv}}(\|\cdot\|_{\infty\wedge 0}) \leq \overline{\operatorname{conv}}(\|\cdot\|_{C\wedge 0}) \leq \overline{\operatorname{conv}}(\|\cdot\|_{1\wedge 0}),$$

hence $\overline{\operatorname{conv}}(\|\cdot\|_{C\wedge 0}) = \|\cdot\|_1$ on the ℓ_1 norm unit ball $\{\|\cdot\|_1 \leq 1\}$. Moreover, the first inequality above requires $\|\cdot\|_1 \leq \overline{\operatorname{conv}}(\|\cdot\|_{C\wedge 0}) < \infty$ on the set $(\operatorname{conv} C)\backslash\{\|\cdot\|_1 \leq 1\}$, but the particular form depends on the shape of $\operatorname{conv} C$.

Let $C = \{\|\cdot\|_p \leq 1\}$. For $0 < p \leq 1$, we get exactly again (A.14), while the case $1 < p < \infty$ is more involved: the conjugate turns out to be

$$\max_{k=1,\ldots,m} (\|\mathbf{w}\| - k)_+, \tag{A.15}$$

where the norm $\|\mathbf{w}\| = (\sum_{i=1}^k |w|_{[i]}^q)^{1/q}$ is the ℓ_q norm of the largest k magnitudes in $\mathbf{w}$. The biconjugate is given by

$$\inf\left\{\sum_{k=1}^m k\lambda_k : \mathbf{w} = \sum_{k=1}^m \lambda_k \mathbf{w}_k, \|\mathbf{w}_k\|_\circ \leq 1\right\}. \tag{A.16}$$

Note that the biconjugate coincides with the cardinality function at the boundary $\{\|\cdot\|_p = 1\}$ and the origin, but at no other points in C, see Figure 1.3 in Chapter 1. Indeed, for any point $\mathbf{w}$ at the boundary, the restricted cardinality function is subdifferentiable: We claim that for $\alpha > 0$ sufficiently large we have for any $\mathbf{x} \in C$, $\|\mathbf{x}\|_0 \geq \|\mathbf{w}\|_0 + \langle \alpha\mathbf{z}, \mathbf{x} - \mathbf{w}\rangle$ where $\mathbf{z} \neq 0$ is chosen to satisfy $\langle \mathbf{z}, \mathbf{w}\rangle = \|\mathbf{z}\|_q \|\mathbf{w}\|_p$. This is verified by assuming first $\mathbf{x}$ and $\mathbf{w}$ share the same indexes for nonzero entries, in which case we have $\langle \alpha\mathbf{z}, \mathbf{x} - \mathbf{w}\rangle \leq 0, \|\mathbf{x}\|_0 = \|\mathbf{w}\|_0$; the other case is verified by letting α be sufficiently large (since $\mathbf{x}$ must have "missed" to match at least one nonzero component of $\mathbf{z}$). It is clear that the biconjugate (and the cardinality function) is subdifferentiable at $\mathbf{0}$, the global minimizer. For any other point $\mathbf{x}$ and its potential subgradient $\mathbf{g}$, the inequality $\|\mathbf{y}\|_0 \geq \|\mathbf{x}\|_0 + \langle \mathbf{g}, \mathbf{y} - \mathbf{x}\rangle$ can not hold for all $\mathbf{y} \in C$. Simply let $\mathbf{y} = 0$ and $\mathbf{y} = (1 \pm \epsilon)\mathbf{x}$ to argue $\langle \mathbf{g}, \mathbf{x}\rangle = 0$; then let $\mathbf{y}$ range over all cardinality 1 vectors and argue $\mathbf{g} = \mathbf{0}$; finally take $\mathbf{y} = \mathbf{0}$ to arrive at a contradiction. Therefore the restricted cardinality function is subdifferentiable at and only at the boundary and the origin. According to Zălinescu (2002, Theorem 2.4.1), these points are precisely where the biconjugate conincides with the original function. We note that in a recent paper Argyriou et al. (2012) derived the convex hull of $\{\mathbf{w} \in \mathbb{R}^m : \|\mathbf{w}\|_0 \leq k, \|\mathbf{w}\|_2 \leq 1\}$, which then induces a norm (through the gauge function, see **??**) that in some sense resembles (A.16).

Let us finally turn back to (A.11), where we truncated the cardinality function rather arbitrarily by the *unit* ℓ_∞ norm ball. Had we known the scale of the optimal solution $\mathbf{w}^\star$, we would be able

to come up with a tighter convex approximation, by tailoring the ℓ_∞ ball accordingly. For instance, if we knew $|w_i^\star| \leq s_i$, then we should truncate the cardinality function by the rescaled ℓ_∞ ball: $\mathsf{B} := \{\mathbf{w} \in \mathbb{R}^m : \forall i, |w_i| \leq s_i\} = \{\mathbf{w} \in \mathbb{R}^m : \forall i, |w_i/s_i| \leq 1\}$. A direct application of Lemma A.6 yields the reweighted closed convex hull

$$\sum_{i=1}^{m} \frac{|w_i|}{s_i} + \iota_{\mathsf{B}}(\mathbf{w}). \tag{A.17}$$

Of course, one usually does not know the optimal scales s_i, but they can be estimated adaptively and iteratively: Fix some s_i (say, 1), use (A.17) as the relaxed regularizer to solve $\mathbf{w}$; then set $s_i = |w_i|$ and iterate. This is precisely the main idea behind the adaptive LASSO of Zou (2006).

A.3 Example 2: Low Rank

In some situations, the parameters we are interested in are presented naturally in a matrix form, and it is not uncommon that this matrix is of low rank or can be well approximated by low rank matrices. To exploit this prior structural information, we could incorporate the rank function as a regularizer in the learning algorithm. Unfortunately, the rank function, being highly nonconvex, is hard to minimize (for some exceptional cases, see Yu and Schuurmans (2011)). On the other hand, it is intuitively clear that the rank is merely a matrix version of the cardinality function that we saw in the previous section, therefore it is conceivable that we can use the same idea to derive the convex hull of the rank function. In fact, it is possible to directly translate any vector result to the matrix domain.

Let $\mathbb{S}^m$ be the vector space of all real symmetric $m \times m$ matrices, and consider the function $F : \mathbb{S}^m \to \mathbb{R} \cup \{\infty\}$. Following Borwein and Lewis (2005), for those functions that only depend on the eigenvalues of their input, we call them spectral functions. There is a natural one-one correspondence between the permutation-invariant function[2] f defined on $\mathbb{R}^m$ and the spectral function F defined on $\mathbb{S}^m$: Indeed, given f, construct $F(W) := f(\mathbf{w})$, where $\mathbf{w}$ constitute the eigenvalues of W; while given F, define $f(\mathbf{w}) := F(\mathrm{Diag}(\mathbf{w}))$, where Diag is the usual operator that turns a m-vector into the corresponding $m \times m$ diagonal matrix. The one-one and onto property of the map that sends f to F is easily verified. The wonderful part of this natural correspondence is that all results about permutation-invariant functions can be trivially translated to those about spectral functions. Thanks to von Neumann's trace inequality (von Neumann 1937), this correspondence also works nicely with the Fenchel conjugate in the sense that if $f \Leftrightarrow F$, then accordingly $f^* \Leftrightarrow F^*$ and $\overline{\mathrm{conv}} f \Leftrightarrow \overline{\mathrm{conv}} F$. More generally, there is a one-one correspondence between symmetric functions[3] and unitarily invariant functions[4], as can be inferred from von Neumann's seminal paper (von Neumann 1937).

[2] A function f is permutation-invariant if $f(\mathbf{w}) = f(P\mathbf{w})$ for any permutation matrix P.
[3] A function f is symmetric if $f(\mathbf{w}) = f(|P\mathbf{w}|)$ for all permutation P, where $|\cdot|$ is the component-wise absolute value.
[4] A function F is unitarily invariant if $F(W) = F(UWV)$ for all unitary matrices U and V.

Now we can apply the reduction. Clearly the rank function is unitarily invariant and corresponds to the cardinality function. Therefore all convex relaxations for the cardinality function immediately yields corresponding convex relaxations for the rank function. In particular, if we relax the cardinality to the ℓ_1 norm, we can similarly relax the rank with the trace norm (sum of all singular values). The effectiveness of the latter relaxation has been rigorously confirmed in Candès and Recht (2009) and many subsequent work. Clearly, all results in Appendix A.2 directly translate to the matrix setting. For some experiments which employed the trace norm to encourage low-rank solutions, see Chapter 4.

A.4 Example 3: Dimensionality Reduction

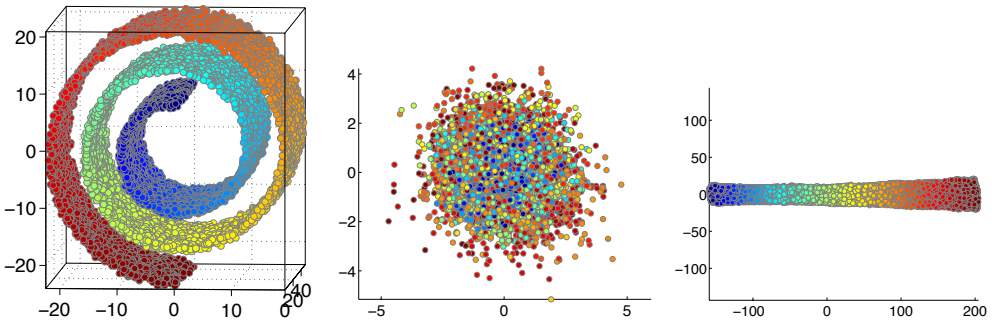
Dimensionality reduction is an ubiquitous and important form of data analysis. Recovering the inherent manifold structure of data—i.e. the local directions of large versus small variation—enables useful representations based on encoding highly varying directions. Not only can this reveal important structure in data, and hence support visualization, it also provides an automated form of noise removal and data normalization that can aide subsequent data analysis.

More specifically, we are given some points $\mathbf{x}_i, i = 1, \ldots, n$, in some high dimensional space $\mathbb{R}^m$, and we want to reduce them to some lower dimensional space $\mathbb{R}^d$ where $d \ll m$, so that some inherent structure hidden in the data is nevertheless preserved. A key assumption is that the given data is sampled from some manifold with low intrinsic dimension. Since by definition a manifold is a locally Euclidean topological space, it makes sense to respect local distances but allow distortions of global distances. Figure A.1(a) shows an example where we sampled 1000 points from the manifold known as the Swiss roll, which is intrinsically two dimensional but embedded in a three dimensional Euclidean space.

Since the seminal work of Roweis and Saul (2000); Tenenbaum et al. (2000), most dimensionality reduction methods can be treated as learning a Gram matrix[5] from data while respecting local distances (Ham et al. 2004). The role of reducing dimensionality is then played by imposing some low rank constraint on the Gram matrix. Quite interestingly, if we follow the previous section naively to relax the rank function in this case to the trace (since the Gram matrix is positive semidefinite), we get a disastrous result such as the one shown in Figure A.1(b). On the other hand, it is known that, to the contrary, if we *maximize* the trace (Weinberger and Saul 2006), which seems to contradict the goal of reducing dimension, we indeed get good results, see Figure A.1(c)!

The explanation turns out to be quite simple: In dimensionality reduction, we not only care about reducing the dimension and retaining local distances, in some sense we also want to stretch the points

[5] That is, the symmetric positive semidefinite matrix $X \in \mathbb{S}^n_+$ with $X_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.



(a) Swiss roll. (b) Minimize trace. (c) Maximize trace.

to increase variation. Thus a more reasonable regularizer for dimensionality reduction appears to be

$$r(X) := -\sum_{i=1}^{d} \sigma_i(X) + \lambda \cdot \sum_{i=d+1}^{n} \sigma_i(X), \tag{A.18}$$

where $\sigma_i(X)$ is the i-th largest eigenvalue value of the positive semidefinite (Gram) matrix X. When minimizing the regularizer (A.18), the second term expresses our desire to reduce dimension by pushing the small eigenvalues to 0 while the first term encourages "stretch out" in the reduced d dimensions, and $\lambda \geq 0$ balances the two different goals. Indeed, this regularizer has been proposed before in Shaw and Jebara (2007), although there only some local alternating algorithm was tried.

It is clear that when $d \in \{0, m\}$, (A.18) is convex; but in all other cases it can be shown non-convex. The latter fact motivates us to derive its closed convex hull. Note that as commented in the previous section, we could have reduced everything to the vector domain. However, since a direct treatment is not any harder, we stick to the matrices.

We first derive the Fenchel conjugate (for $d \geq 1$):

$$\begin{aligned} r^*(Y) &= \sup_{X \succeq 0} \langle X, Y \rangle - r(X) \\ &= \sup_{X \succeq 0} \sum_{i=1}^{d} \sigma_i(X)(\sigma_i(Y) + 1) + \sum_{i=d+1}^{n} \sigma_i(X)(\sigma_i(Y) - \lambda) \\ &= \iota_{\{\sigma_1(\cdot) \leq -1\}}(Y). \end{aligned}$$

Next we derive the biconjugate (closed convex hull):

$$\begin{aligned} r^{**}(Z) &= \sup_{Y} \langle Y, Z \rangle - r^*(Y) \\ &= \sup_{Y} \langle Y, Z \rangle - \iota_{\{\sigma_1(\cdot) \leq -1\}}(Y) \\ &= \iota_{\mathbb{S}_+^n}(Z) - \sum_{i=1}^{n} \sigma_i(Z). \end{aligned}$$

To summarize, the closed convex hull of (A.18) is

$$(\overline{\text{conv}} r)(X) = \begin{cases} \lambda \cdot \sum_{i=1}^{n} \sigma_i(X), & d = 0 \\ -\sum_{i=1}^{n} \sigma_i(X), & d \in \{1, \ldots, n\}, \end{cases} . \tag{A.19}$$

This result is a little surprising and does not seem to have been observed previously. It also partially explains why maximizing the trace, opposed to the conventional wisdom that minimizing the trace leads to low rank, is more effective in dimensionality reduction.

Note that for any $d \geq 1$, we get essentially the same closed convex hull for the regularizer (A.18). Computationally this is convenient but theoretically it is inferior since any returned solution is not customized for any targeted dimension d. This problem can be fixed by adding some extra regularization to (A.18). We refer the details to Yu et al. (2012).