Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA ”TULLIO LEVI-CIVITA”
Corso di Laurea Magistrale in Matematica

Tesi di laurea magistrale

Proximal algorithms for nuclear norm system identification

Candidato:
Chiara Faccio
Matricola 1157072

Relatore:
Prof. Fabio Marcuzzi

Anno Accademico 2018 - 2019
19 Aprile 2019

Contents

Introduction

System Identification concerns the estimation of dynamical systems models through input and output measurements. In recent years, numerous papers have been published in this subject and, recently, a new method based on singular value decompositions (SVD) has emerged, the so-called subspace methods. However, the latter does not allow to add regularization terms to the problem or to extend it to the case in which part of the inputs or outputs is missing, for example, due to the breakage of a sensor. For this reason, a convex optimization formulation based on the nuclear norm penalty offers an interesting alternative. It promotes a low rank optimum: in fact, the nuclear norm of a matrix, by definition, is the sum of its singular values, therefore we can interpret the nuclear norm as a sort of convex relaxation of the rank. Then, we must solve the following convex optimization problem:

$$\min_{y}\left\{\|A(y)\|_{*}+\frac{1}{2}\|y-y_{meas}\|_{2}\right\} \tag{1}$$

where y_{meas} are the measured outputs, A is a linear mapping and $\|\cdot\|_{*}$ indicates the nuclear norm. This optimization problem is used as a preprocessing step: it computes a modified output sequence which is passed to the standard subspace method. In this way, it regularizes the outputs or it reconstructs the missing data.

As a result, the interest in convex optimization techniques has recently increased [22] [24], including the *proximal algorithms* that we have studied in this thesis. They are algorithms for solving a convex optimization problem which uses a proximal operator of the objective terms: let $f:\mathbb{R}^n \rightarrow \mathbb{R}\cup +\infty$ be a closed proper function, the *proximal operator* $prox_{\lambda f}:\mathbb{R}^n \rightarrow \mathbb{R}^n$ of f is defined by

$$prox_{\lambda f}(v)=\operatorname*{argmin}_{x}\left\{f(x)+\frac{1}{2\lambda}\|x-v\|_{2}^{2}\right\}$$

In the course of this thesis, we will see 4 different proximal algorithms to solve problems of the type $\min\{f(x)+g(x)\}$ where, usually, f is differentiable and g is nonsmooth :

- Proximal Gradient Method: it combines a gradient type iterative method with the proximal operator;
- Accelerated Proximal Gradient method: it is an accelerated version of the PGM;
- Alternating Direction Method of Multipliers: this method calculates the proximal operator separately on f and on g and then it combines them into a dual variable;
- Proximal Newton Method : it combines the Newton method with the proximal operator.

In some applications, it is advantageous to apply a proximal algorithm to the dual problem, instead of the primal one. Given the problem $\min_{x\in\mathbb{R}^n}\{f(x)+g(Ax)\}$, with $A:\mathbb{R}^n\to\mathbb{R}^{p\times q}$ a linear mapping, its dual formulation is

$$\min_{u\in\mathbb{R}^{p\times q}}\{f^*(-A_{adj}u)+g^*(u)\},$$

where with f^* we indicate the convex conjugate of the function f and $A_{adj}:\mathbb{R}^{p\times q}\to\mathbb{R}^n$ is the adjoint mapping of A.

The use of PGM, and of its accelerated version to solve a nuclear norm optimization problem, had already been proposed in the article of Fazel, Pong, Sun, Tseng [12], where, to derive the dual formulation of the problem, the authors relied on the minimization/maximization of the Lagrangian. Instead, in this thesis, we derive the dual formulation directly from the use of convex conjugates of the functions. Another difference with the article is that we solve a more general nuclear norm problem: instead of considering the Euclidean norm in the second part of (1), we study $(y-y_{meas})^T E(y-y_{meas})$, with E a positive semidefinite matrix. However, although this result is interesting from a theoretical point of view, it hasn't shown better performances, at this moment.

Indeed, as it will emerge from the experiments, we are not interested in finding the minimum of the problem (1) in an accurate way, since if we find a nuclear norm too small we risk to underestimate the order of the model; on the other side, stopping at a high nuclear norm we risk to overestimate it. As a consequence, we will see how the results obtained through ADMM and FPGM, and in some cases PGM, are slightly better than the other one, since PNM goes quickly to a minimum.

Another contribution of this thesis is related in particular to model order selection. In the literature of subspace methods, and consequently in that of nuclear norm minimization, the authors use a threshold on singular values of some block-Hankel matrices to determine the model order. Here we show that this brings to choose larger models than necessary, with well known disadvantages, e.g. over-fitting. Instead, here we propose to use the parsimony principles, that in this thesis has been incorporated in the proximal algorithms and tested experimentally. The results show that the modified ADMM algorithm here proposed obtains a better fit with a lower order model in a wide set of real data. The experiments with simulation data confirm this behavior systematically. The results does not change when using different proximal algorithms.

The thesis is organized in the following way: in Chapter 1 it is explained in detail what a proximal operator is and what its properties are. In Chapter 2 the proximal algorithms are studied in detail. For each of them, we see the convergence rate and their interpretation, since they can be interpreted as generalizations of other known algorithms. In Chapter 3 the operation of the subspace method and the formation of the nuclear norm optimization problem is briefly resumed. Later, we see in detail how to apply the various algorithms to the problem. In Chapter 4 the numerical experiments are presented.

Chapter 1

Proximal operator

1.1 Definition

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function, which means that its epigraph is a nonempty closed convex set. We remember that the epigraph of a function f is

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

Instead, the effective domain of f is the set of points for which f takes on finite values:

$$dom(f) = \{x \in \mathbb{R}^n : f(x) < +\infty\}$$

Assuming these hypotheses, we can define the proximal operator:

Definition 1. *The* ***proximal operator*** $prox_{\lambda f} : \mathbb{R}^n \to \mathbb{R}^n$ *of* f *is defined by*

$$prox_{\lambda f}(v) := \underset{x}{\operatorname{argmin}} \big\{ f(x) + \frac{1}{2\lambda} \|x - v\|_2^2 \big\},$$

where $\|\cdot\|_2$ *is the usual Euclidean norm and the parameter* $\lambda > 0$*.*

We notice that the function minimized on the righthand size is closed and strongly convex because it is a sum of the closed and strongly convex function $\frac{1}{2}\|\cdot - x\|_2^2$ and the closed and convex function f; so it has an unique minimizer for every $v \in \mathbb{R}^n$.

The mapping $prox_{\lambda f}$ takes a point $v \in \mathbb{R}^n$ and moves it. The points in the domain of the function stay in the domain and move towards the minimum of the function, while the points outside the domain move to the boundary and towards the minimum of the function. The parameter λ controls the extent to which the proximal operator maps points towards the minimum: with larger values of λ associated with mapped points near the minimum, and smaller values giving smaller movement towards the minimum. So $prox_{\lambda f}(v)$ is a point that compromises between minimizing f and being near to v.

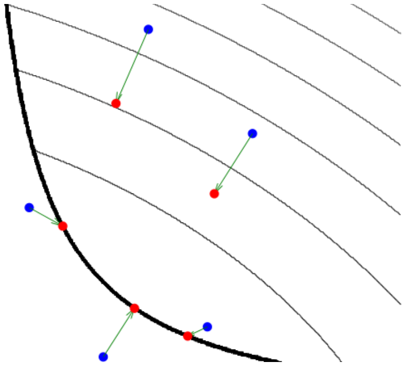


Figure 1.1: Evaluating a proximal operator

In Figure 1.1 the thin black lines are the level curves of f, the thicker black line indicates the boundary of the function and evaluating the proximal operator at the blue points moves them to the corresponding red points.

Example 1. *If f is the indicator function*

$$I_{\mathcal{C}}(x) = \begin{cases} 0, & x \in \mathcal{C} \\ +\infty, & x \notin \mathcal{C} \end{cases}$$

where $\mathcal{C}$ is a closed nonempty convex set, the proximal operator of f reduces to Euclidean projection onto $\mathcal{C}$. So in this case, the proximal operator can be interpreted as generalized projection.

1.2 Properties

Now we discuss the main properties of proximal operators, which are used, for example, to establish convergence of a proximal algorithm or to derive a method for evaluating the proximal operator.

- **separable sum**: if f is separable across two variables, i.e. $f(x,y) = \varphi(x) + \psi(y)$, then

  $$prox_f(v,w) = (prox_\varphi(v), prox_\psi(w)).$$

  So evaluating the proximal operator of a separable function reduces to evaluating the proximal operators for each of the separable parts, which can be done independently. In tis way if f is fully separable, $f(x) = \sum_{i=1}^n f_i(x_i)$, then $(prox_f(v))_i = prox_{f_i}(v_i)$

- **fixed points**: the point x^* minimizes f if and only if x^* is a fixed point of $prox_f$.

  Proof. Without loss of generality we can consider $\lambda = 1$, in fact x^* minimizes f if and only if x^* minimizes λf. If x^* minimizes f, $f(x) \geq f(x^*)$ $\forall x$, then

  $$f(x) + \frac{1}{2}\|x - x^*\|^2 \geq f(x^*) = f(x^*) + \frac{1}{2}\|x^* - x^*\|^2$$

  So x^* minimizes $f(x) + \frac{1}{2}\|x - x^*\|^2$. It follows that $x^* = prox_f(x^*)$.

  We prove the converse, $\hat{x} = prox_f(v)$ if and only if $0 \in \partial f(\hat{x}) + (\hat{x} - v)$, where $\partial f(x)$ is the subdifferential of f at x, defined by

  $$\partial f(x) = \{y : f(z) \geq f(x) + y^t(z - x), \forall z \in dom(f)\} \tag{1.1}$$

  When f is differentiable, we have that $\partial f(x) = \{\nabla f(x)\}$, $\forall x$.

  Taking $\hat{x} = v = x^*$, we have that $0 \in \partial f(x^*)$ and so x^* minimizes f.

  □

  Since minimizers of f are fixed points of $prox_f$, we can minimize f by finding a fixed point of its proximal operator.

- **Moreau decomposition**:

  the following relation always holds:

  $$v = prox_{f^*}(v) + prox_f(v)$$

  where $f^*(y) = \sup_x(y^t x - f(x))$ is the conjugate convex of f.

Proof. let $v \in \mathbb{R}^n$ and denote $u = prox_f(v)$. Then

$$0 \in \partial f(u) + (u - v) \Rightarrow v - u \in \partial f(u).$$

Using a property of the conjugate, that we will see later, we have that $u \in \partial f^*(v - u)$.

Then

$$v - u = prox_{f^*}(v) \quad \Rightarrow \quad v = u + prox_{f^*}(v) = prox_f(v) + prox_{f^*}(v)$$

□

The general Moreau decomposition is

$$v = prox_{\lambda f^*}(v) + \lambda prox_{\frac{1}{\lambda}f}(\frac{v}{\lambda}).$$

- **relation with $\boldsymbol{\partial f}$**: the proximal operator $prox_{\lambda f}$ and the subdifferential operator ∂f, defined in 1.1, are related as follows:

$$prox_{\lambda f} = (I + \lambda \partial f)^{-1} \tag{1.2}$$

Proof. $z \in (I+\lambda\partial f)^{-1}(x) \iff x \in (I+\lambda\partial f)(z) \iff x \in z+\lambda\partial f(z)$ $\iff 0 \in \partial f(z) + \frac{1}{\lambda}(z - x) \iff 0 \in \partial_z\Big(f(z) + \frac{1}{2\lambda}\|z - x\|^2\Big)$.

Now the function $f(z) + \frac{1}{2\lambda}\|z - x\|^2$ is strongly convex, so there is an unique minimum. Then

$$z = \underset{u}{\operatorname{argmin}}\{f(u) + \frac{1}{2\lambda}\|u - x\|^2\} \iff z = prox_{\lambda f}(x).$$

□

1.3 Proximal algorithms

A proximal algorithm is an algorithm for solving a convex optimization problem which uses the proximal operators of the objective terms. The properties of $prox_f$ above suggest several potential perspectives on this algorithm, such as a fixed point iteration or the possibility to pass to a dual problem.

There are many reasons to use the proximal algorithms : they work under extremely general conditions, including cases where the functions are nonsmooth; they can be fast since there can be simple proximal operators for functions that are otherwise challenging to handle in an optimization problem. Then they can be used to solve many problems and they are easy, because they can be interpreted as generalizations of other algorithms.

We describe, briefly, some important proximal algorithms for solving convex optimization problems:

- **proximal minimization**:

  $$x^{k+1} := prox_{\lambda f}(x^k)$$

  where $f : \mathbb{R}^n \to \mathbb{R} \cup +\infty$ is a closed proper function, k is the iteration counter and x^k denotes the k-th iterate of the algorithm. In this way if f has a minimum, x^k converges to the set of minimizers of f. A variation on the algorithm uses parameter values that change in each iteration. However, this basic method has not found many applications.

- **proximal gradient method**: this method, and the other ones, will be studied in detail in the next chapter, so here we present briefly the algorithm. Consider the problem

  $$minimize \quad \{f(x) + g(x)\}$$

  where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R} \cup +\{\infty\}$ are closed proper convex functions, f is differentiable and g can be nonsmooth. The proximal gradient method is

  $$x^{k+1} := prox_{\lambda g}(x^k - \lambda \nabla f(x^k))$$

- **accelerated proximal gradient method**: it is an accelerated version of the proximal gradient algorithm. A version is

$$\begin{aligned} y^{k+1} &:= x^k + t^k(x^k - x^{k-1}) \\ x^{k+1} &:= prox_{\lambda g}(y^{k+1} - \lambda \nabla f(y^{k+1})) \end{aligned}$$

  where $t^k \in [0, 1)$ is an extrapolation parameter and λ is the usual step size. These parameters must be chosen in specific ways to achieve the convergence acceleration.

- **alternating direction method of multipliers**: considered the problem

$$minimize \quad \{f(x) + g(x)\}$$

  where $f, g : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\{\infty\}$ are closed proper convex functions and both f and g can be nonsmooth. The method, also known as Douglas-Rachford splitting, is

$$\begin{aligned} x^{k+1} &:= prox_{\lambda f}(z^k - u^k) \\ z^{k+1} &:= prox_{\lambda g}(x^{k+1} + u^k) \\ u^{k+1} &:= u^k + x^{k+1} - z^{k+1} \end{aligned}$$

  The advantage of this method is that the objective terms are handled separately, so it is useful when the proximal operators of f and g are efficiently evaluated, but the proximal operator of $f + g$ is not easy to compute.

- **proximal Newton method**: considered the problem

$$minimize \quad \{f(x) + g(x)\}$$

  where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a proper convex, continuosly differentiable function and its gradient is Lipshitz continuos; instead $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is a proper convex, but not necessarily differentiable function, whose proximal mapping can be evaluated efficiently. We present a line search algorithm, where first we choose a descent direction Δx^k , then we find a step size t^k and, in the end, we update the iterate x^k:

$$\begin{aligned}&\Delta x^k = prox_g^H(x^{k-1} - H^{-1}\nabla f(x^{k-1})) - x^{k-1}\\&x^k = x^{k-1} + t^k\Delta x^k\end{aligned}$$

where H is the hessian of the function f or its approximation.

Chapter 2

Proximal algorithms

2.1 Proximal gradient method

In this section we describe a very popular algorithm to solve

$$minimize \quad \{\varphi(x) = f(x) + g(x)\} \tag{2.1}$$

The following assumptions are made throughout the section:

- $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth closed proper convex function, differentiable and with Lipschitz continuos gradient, i.e.

  $$\|\nabla f(x) - \nabla f(y)\| \leq \mathcal{L}_f \|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

  where $\mathcal{L}_f > 0$;

- $g : \mathbb{R}^n \to \mathbb{R}$ is a continuous, closed proper convex function, it can be nonsmooth and it has an efficient proximal mapping;

- the problem 2.1 is solvable, i.e. $\exists x^* = \operatorname{argmin} \varphi(x) \neq \emptyset$.

We split the objective into two terms, one of which is differentiable. This splitting is not unique and different partitions lead to different implementations of the PGM (proximal gradient method).

Example 2. *The lasso problem is*

$$minimize \quad \frac{1}{2}\|Ax - b\|_2^2 + \gamma\|x\|_1$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $\gamma > 0$. The problem can be interpreted as finding a sparse solution to a least squares. So we can consider the splitting

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 \qquad g(x) = \gamma\|x\|_1$$

Despite its simplicity, problem2.1 encompasses a large variety of applications:

Example 3. *The constrained optimization problem can be formulated as 2.1: in fact, if we have the problem of minimizing f subject to the constraint $x \in \mathcal{C}$, with $\mathcal{C}$ a nonempty set, we set g the indicator function of $\mathcal{C}$.*

The presence of a nonsmooth function prevents from applying classical optimization algorithms such as gradient descent. In fact, these methods are based on derivatives and do not apply to the minimization of non-differentiable functions. Consequently, one way to deal with this kind of problems is through the PGM: it combines the gradient descent with the proximal mapping.

$$x^{k+1} := prox_{\lambda g}(x^k - \lambda^k \nabla f(x^k))$$

where k is an iteration counter and x^0 is an initial value in $\mathbb{R}^n$. The parameter $\lambda^k > 0$ is a step size. It can be fixed and we will prove that if $\lambda = \lambda^k \in (0, 1/\mathcal{L}_f]$ the method will converge. If $\mathcal{L}_f$ is not known, λ^k can be found by a line search. An example of line search is the following:

```
given $x^k$, $\lambda^{k-1}$, and parameter $\alpha \in (0,1)$
let $\lambda = \lambda^{k-1}$
repeat
    1. let $z = prox_{\lambda g}(x^k - \lambda \nabla f(x^k))$
    2. break if  $f(z) \leq \hat{f}_\lambda(z, x^k)$
    3. update $\lambda = \alpha\lambda$
return $\lambda^k = \lambda$, $x^{k+1} = z$.
```

A typical value for the line search parameter α is 1/2. The function $\hat{f}_\lambda$ is easy to evaluate:

$$\hat{f}_\lambda(x, y) = f(y) + \langle x - y, \nabla f(y) \rangle + \frac{1}{2\lambda}\|x - y\|^2 \tag{2.2}$$

Lemma 1. *let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuosly differentiable function with Lipschitz continuos gradient and Lipschitz constant $\mathcal{L}_f$. Then for any $L \geq \mathcal{L}_f$*

$$f(x) \leq f(y) + \langle x-y, \nabla f(y) \rangle + \frac{L}{2}\|x-y\|^2 \quad \forall x, y \in \mathbb{R}^n$$

Proof. We consider the function $g(t) = f(y + t(x-y))$, so $g(1) = f(x)$, $g(0) = f(y)$, $\frac{\partial g}{\partial t} = \langle x-y, \nabla f(y + t(x-y)) \rangle$:

$$\begin{aligned}
f(x) - f(y) = g(1) - g(0) &= \int_0^1 \frac{\partial g}{\partial t}\, dt = \int_0^1 \langle x-y, \nabla f(y + t(x-y)) \rangle \, dt \\
&\leq \int_0^1 \langle x-y, \nabla f(y) \rangle \, dt + \mid \int_0^1 \langle x-y, (\nabla f(y + t(x-y)) - \nabla f(y)) \rangle \, dt \mid \\
&\leq \langle x-y, \nabla f(y) \rangle + \|x-y\| \int_0^1 \mathcal{L}_f t \|x-y\| \, dt \\
&= \langle x-y, \nabla f(y) \rangle + \frac{\mathcal{L}_f}{2}\|x-y\|^2 \leq \langle x-y, \nabla f(y) \rangle + \frac{L}{2}\|x-y\|^2
\end{aligned}$$

□

Using Lemma 1 we have:

$$f(x) - f(y) \leq \langle x-y, \nabla f(y) \rangle + \frac{L}{2}\|x-y\|^2 \leq \langle x-y, \nabla f(y) \rangle + \frac{1}{2\lambda}\|x-y\|^2$$

where the last inequality is true if $\lambda \in (0, 1/\mathcal{L}_f]$. So, it is a convex upper bound of f and satisfies $\hat{f}_\lambda(x, x) = f(x)$ when $\lambda \in (0, 1/\mathcal{L}_f]$.

Now we see a geometric interpretation: the gradient step (forward) moves the iterate x^k towards the minimum of f, while the proximal step (backward) makes progress towards the minimim of g. This alternation will ultimately lead to the minimum of the sum of these two functions. We can see this in Figure 2.1.

There are some special cases:

- $g = I_\mathcal{C} \Rightarrow prox_{\lambda g}$ is projection onto $\mathcal{C} \Rightarrow$ PGM reduces to the projected gradient method;
- $f = 0 \Rightarrow$ PGM reduces to proximal minimization;
- $g = 0 \Rightarrow$ PGM reduces to the standard gradient descent method.

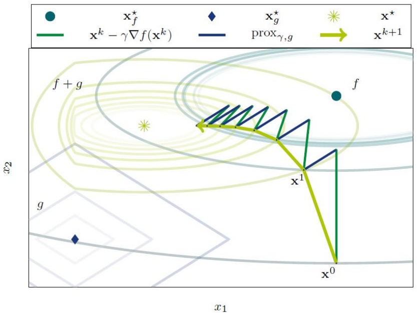


Figure 2.1: Path that the PGM creates to reach the optimal value

2.1.1 Convergence

When $g \equiv 0$, the PGM reduces to the gradient method and the sequence of function values $\varphi(x^k)$ converges to $\varphi(x^*)$ with a sublinear rate: $\varphi(x^k) - \varphi(x^*) \in O(c/k)$, $c > 0$ constant and x^* is the optimal value of problem 2.1. We want to prove that PGM shares the same rate of convergence.

To reduce the notation, we define the PGM with $P_L(y) := prox_{\frac{1}{L}g}(y - \frac{1}{L}\nabla f(y))$, with $L > 0$.

Given $\varphi(x) = f(x) + g(x)$, we consider the following quadratic approximation at a given point y :

$$Q_L(x, y) := f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 + g(x)$$

$Q_L(x, y)$ is a convex function, so it admits a unique minimizer and $P_L(y) = \mathrm{argmin}_x\{Q_L(x, y)\}$.

Lemma 2. *$\forall y \in \mathbb{R}^n$, one has $z = P_L(y) \iff \exists\gamma(y) \in \partial g(z)$, the subdifferential of g (1.1), such that*

$$\nabla f(y) + L(z - y) + \gamma(y) = 0$$

Lemma 3. *Let $y \in \mathbb{R}^n$, $L > 0$ be such that $\varphi(P_L(y)) \leq Q_L(P_L(y), y)$, then* $\forall x \in \mathbb{R}^n$

$$\varphi(x) - \varphi(P_L(y)) \geq \frac{L}{2}\|P_L(y) - y\|^2 + L\langle y - x, P_L(y) - y\rangle.$$

Proof. $\varphi(P_L(y)) \leq Q_L(P_L(y), y) \Rightarrow \underbrace{\varphi(x) - \varphi(P_L(y)) \geq \varphi(x) - Q_L(P_L(y), y)}_{\blacksquare}$

Now f and g are convex for hypothesis, so:
$f(x) \geq f(y) + \langle x - y, \nabla f(y) \rangle$
$g(x) \geq g(P_L(y)) + \langle x - P_L(y), \gamma(y) \rangle$

$$f(x) + g(x) = \varphi(x) \geq f(y) + \langle x - y, \nabla f(y) \rangle + g(P_L(y)) + \langle x - P_L(y), \gamma(y) \rangle$$

By definition of $P_L(y)$:

$$Q_L(P_L(y), y)) = f(y) + \langle P_L(y) - y, \nabla f(y) \rangle + \frac{L}{2} \|P_L(y) - y\|^2 + g(P_L(y))$$

We replace in ■:

$$\begin{aligned}
\varphi(x) - \varphi(P_L(y)) &\geq f(y) + \langle x - y, \nabla f(y) \rangle + g(P_L(y)) + \langle x - P_L(y), \gamma(y) \rangle \\
&\quad - f(y) - \langle P_L(y) - y, \nabla f(y) \rangle - \frac{L}{2} \|P_L(y) - y\|^2 - g(P_L(y)) \\
&= -\frac{L}{2} \|P_L(y) - y\|^2 + \langle x - P_L(y), \nabla f(y) + \gamma(y) \rangle \\
&= \text{ we use lemma 2} \\
&= -\frac{L}{2} \|P_L(y) - y\|^2 + \langle x - P_L(y), L(y - P_L(y)) \rangle \\
&= -\frac{L}{2} \|P_L(y) - y\|^2 + L\langle P_L(y) - x + y - y, P_L(y) - y) \rangle \\
&= -\frac{L}{2} \|P_L(y) - y\|^2 + L\|P_L(y) - y\|^2 + L\langle y - x, P_L(y) - y) \rangle \\
&= \frac{L}{2} \|P_L(y) - y\|^2 + L\langle y - x, P_L(y) - y) \rangle
\end{aligned}$$

□

We note that from Lemma 1 it follows that if $L \geq \mathcal{L}_f$, and so $\lambda \in (0, 1/\mathcal{L}_f)$, the condition $\varphi(P_L(y)) \leq Q(P_L(y), y)$ is always satisfied for $P_L(y)$.

Moreover, it exists α and β such that $\beta \mathcal{L}_f \leq L^k := \frac{1}{\lambda^k} \leq \alpha \mathcal{L}_f$.

Theorem 2. *Let $\{x^k\}_{k \geq 1}$ the sequence generated by PGM, then*

$$\varphi(x^k) - \varphi(x^*) \leq \frac{\alpha \mathcal{L}_f \|x^0 - x^*\|^2}{2k}, \quad \forall k \geq 1.$$

Proof. We use Lemma 3 with $x = x^*$, $y = x^n$, $L = L^{n+1}$:

$$\begin{aligned}\frac{2}{L^{n+1}}(\varphi(x^*) - \varphi(x^{n+1})) &\geq \|x^{n+1} - x^n\|^2 + 2\langle x^n - x^*, x^{n+1} - x^n\rangle \\ &= \langle x^{n+1} - x^n, x^{n+1} - x^n\rangle + 2\langle x^n - x^*, x^{n+1} - x^n\rangle \\ &= \langle x^{n+1} + x^n - 2x^*, x^{n+1} - x^n\rangle \\ &= \|x^* - x^{n+1}\|^2 - \|x^* - x^n\|^2\end{aligned}$$

Using the fact that $\varphi(x^*) - \varphi(x^{n+1} \leq 0$ and $\beta\mathcal{L}_f \leq L^k \leq \alpha\mathcal{L}_f$:

$$\frac{2}{\alpha\mathcal{L}_f}(\varphi(x^*) - \varphi(x^{n+1})) \geq \|x^* - x^{n+1}\|^2 - \|x^* - x^n\|^2$$

Summing this inequality over $n = 0, ..., k-1$ gives:

$$\frac{2}{\alpha\mathcal{L}_f}(k\varphi(x^*) - \sum_{n=0}^{k-1}\varphi(x^{n+1})) \geq \|x^* - x^k\|^2 - \|x^* - x^0\|^2 \tag{2.3}$$

Invoking Lemma 3 with $x = y = x_n$, $L = L^{n+1}$:

$$\frac{2}{L^{n+1}}(\varphi(x^n) - \varphi(x^{n+1})) \geq \|x^{n+1} - x^n\|^2$$

Using the fact that $\varphi(x^n) - \varphi(x^{n+1}) \geq 0$ and $\beta\mathcal{L}_f \leq L^{n+1}$:

$$\frac{2}{\beta\mathcal{L}_f}(\varphi(x^n) - \varphi(x^{n+1})) \geq \|x^{n+1} - x^n\|^2$$

Multiplying by n and summing over $n = 0, ..., k-1$ gives:

$$\frac{2}{\beta\mathcal{L}_f}\sum_{n=0}^{k-1}(n\varphi(x^n) - (n+1)\varphi(x^{n+1}) + \varphi(x^{n+1})) \geq \sum_{n=0}^{k-1} n\|x^{n+1} - x^n\|^2$$

$$\frac{2}{\beta\mathcal{L}_f}(-k\varphi(x^k) + \sum_{n=0}^{k-1}\varphi(x^{n+1})) \geq \sum_{n=0}^{k-1} n\|x^{n+1} - x^n\|^2 \tag{2.4}$$

Summing 2.3 and 2.4 times β/α:

$$\begin{aligned}\frac{2k}{\alpha\mathcal{L}_f}(\varphi(x^*) - \varphi(x^k)) &\geq \|x^* - x^k\|^2 + \frac{\beta}{\alpha}\sum_{n=0}^{k-1} n\|x^{n+1} - x^n\|^2 - \|x^* - x^0\|^2 \\ &\geq -\|x^* - x^0\|^2\end{aligned}$$

Then:

$$\varphi(x^*) - \varphi(x^k) \geq -\frac{\alpha \mathcal{L}_f \|x^0 - x^*\|^2}{2k}$$

$$\varphi(x^k) - \varphi(x^*) \leq \frac{\alpha \mathcal{L}_f \|x^0 - x^*\|^2}{2k}$$

□

2.1.2 Interpretations

The proximal gradient method can be interpreted in many ways.

- **majorization-minimization**: a majorization-minimization algorithm for minimizing a function $\varphi : \mathbb{R}^n \to \mathbb{R}$ consists of the iteration $x^{k+1} = \operatorname{argmin}_x \hat{\varphi}(x, x^k)$, where $\hat{\varphi}(\cdot, x^k)$ is a convex upper bound of φ such that $\hat{\varphi}(x, x^k) \geq \varphi(x)$ and $\hat{\varphi}(x, x) = \varphi(x)$ for all x. The reason for the name is that the algorithm increases (upper bounding) the objective term and then it minimizes the majorization.

  At this point, for an upper bound of f, we consider the function $\hat{f}_\lambda$ described in 2.2. For fixed y the functiom is convex, $\hat{f}_\lambda(x, x) = f(x)$ and it is an upper bound on f when $\lambda \in (0, 1/\mathcal{L}_f]$.

  We define
  $$q_\lambda(x, y) := \hat{f}_\lambda(x, y) + g(x)$$

  It is a surrogate for $f + g$ when $\lambda \in (0, 1/\mathcal{L}_f]$. Then we can prove that

  $$x^{k+1} = \underset{x}{\operatorname{argmin}}\, q_\lambda(x, x^k) \qquad \Longleftrightarrow \qquad x^{k+1} = prox_{\lambda g}(x^k - \lambda \nabla f(x^k))$$

  In fact:

  $$\nabla_x q_\lambda(x, x^k) = \nabla f(x^k) + \frac{1}{\lambda}(x - x^k) + \partial g(x) = 0 \iff$$
  $$x + \lambda \partial g(x) = x^k - \lambda \nabla f(x^k) \iff (I + \lambda \partial g)(x) = x^k - \lambda \nabla f(x^k) \qquad \iff$$
  $$x = (I + \lambda \partial g)^{-1}(x^k - \lambda \nabla f(x^k))$$

  Using the property 1.2, we have $(I + \lambda \partial g)^{-1} = prox_{\lambda g}$, so $x = prox_{\lambda g}(x^k - \lambda \nabla f(x^k)) \implies x^{k+1} = prox_{\lambda g}(x^k - \lambda \nabla f(x^k))$.

- **fixed point iteration** : the PGM can be interpreted also as a fixed point iteration. A point x^* is a solution of $min\{f(x) + g(x)\} \iff 0 \in \nabla f(x^*) + \partial g(x^*)$. Let $\lambda > 0$:

  $$0 \in \lambda \nabla f(x^*) + \lambda \partial g(x^*) \iff 0 \in \lambda \nabla f(x^*) - x^* + x^* + \lambda \partial g(x^*)$$

  $$x^* - \lambda \nabla f(x^*) \in x^* + \partial g(x^*) \iff (I - \lambda \nabla f)(x^*) \in (I + \lambda \partial g)(x^*)$$

  $$x^* = (I + \lambda \partial g)^{-1}(I - \lambda \nabla f)(x^*) \iff x^* = prox_{\lambda g}(x^* - \lambda \nabla f(x^*))$$

  The last two expressions hold with equality and not just containment because the proximal operator is single-valued.

  In this way x^* minimizes $f + g$ if and only if x^* is a fixed point of the forward-backward operator $(I+\lambda \partial g)^{-1}(I-\lambda \nabla f)$. The PGM repeatedly applies this operator to obtain a fixed point and, thus, a solution to the original problem.

  The condition $\lambda \in (0, 1/\mathcal{L}_f]$ guarantees that the forward-backward operator is averaged and thus the iteration converges to a fixed point, when one exists.

- **forward-backward integration of gradient flow**: the PGM can be interpreted using gradient flows, which take the form :

  $$\frac{d}{dt}x(t) = -\nabla f(x(t)) - \nabla g(x(t)) \tag{2.5}$$

  assuming that also g is differentiable. To obtain a discretization, we replace the derivate with

  $$\frac{d}{dt}x(t) \approx \frac{x^{k+1} - x^k}{h}$$

  We replace also the value $x(t)$ on the righthand side of 2.5with either x^k (giving the forward Euler discretization) or x^{k+1} (giving the backward Euler discretization).Since the PGM is a forward-backward algorithm, we use both x^k and x^{k+1} on the righthand side.The result is:

  $$\frac{x^{k+1} - x^k}{h} = -\nabla f(x^k) - \nabla g(x^{k+1}) \implies x^{k+1} + h \nabla g(x^{k+1}) = x^k - h \nabla f(x^k)$$

$$\implies x^{k+1} = (I + h\nabla g)^{-1}(I - h\nabla f)x^k$$

It is the proximal gradient iteration when $h = \lambda$. So PGM can be interpreted as a method for numerically integrating the gradient flow differential equation that uses a forward Euler step for the differentiable part f and a backward Euler step for the part g.

2.2 Fast proximal gradient method

It exists an accelerated version of PGM. In addition to the original iterate x^k, FPGM computes an extrapolated sequence y^k. The basic version is:

```
$y^1 = x^0 \quad t_1 = 1$
for $k \geq 1$ :
    $x^k = prox_{\lambda g}(y^k - \lambda \nabla f(y^k))$
    $t_{k+1} = \frac{1 + \sqrt{1 + 4{t_k}^2}}{2}$
    $y^{k+1} = x^k + \frac{t_k - 1}{t_{k+1}}(x^k - x^{k-1})$
```

Like PGM, $\lambda \in (0, 1/\mathcal{L}_f)$, but if $\mathcal{L}_f$ is not know, the step size λ^k can be found by a line search similar to the line search in PGM.

The main difference between PGM and FPGM is that the operator $prox_{\lambda g}(\cdot - \lambda\nabla f(\cdot))$ is not employed to the previous point x^{k-1}, but at the point y^k, which uses a linear combination of the previous two points x^{k-2} and x^{k-1}.

We notice that $w^k := \frac{t_k - 1}{t_{k+1}} \in [0, 1)$.

We remember that the **extrapolation** is the process of taking data values at points $x_1, ..., x_n$ and approximating a value outside the range of the given points. To do this, we need informations about the model (linear, quadratic,...). So, in the case of FPGM, y_k produce a "corrective" movement, in fact with the PGM at the later steps the gradient becomes smaller and so the progresses are slow. Instead, with the FPGM if the iterate is near the solution, it is going to continue pushing in that direction and it improves when the gradient is small. However with this approach, we are not sure that $\varphi(x^k) \leq \varphi(x^{k-1})$, in fact some oscillations can be formed, called Nesterov ripples, but they are also necesssary to achieve a faster overall convergence.

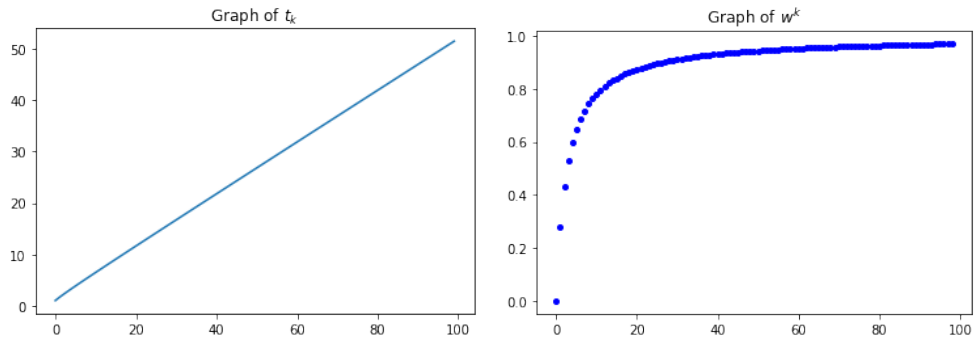


Figure 2.2: The behavior of $\{t_k\}_k$ and $\{w^k\}_k$.

2.2.1 Convergence

In the convex case the iterates of FPGM satisfy $\varphi(x^k) - \varphi(x^*) \in O(1/k^2)$, where x^* is the optimal value of the problem 2.1. To prove this, we introduce some lemmas:

Lemma 4. *the sequences $\{x^k, y^k\}_{k\geq 1}$ generated via FPGM satisfy for every* $k \geq 0$

$$2\lambda^k {t_k}^2 v_k - 2\lambda^{k+1} {t_{k+1}}^2 v_{k+1} \geq \|u_{k+1}\|^2 - \|u_k\|^2$$

where $v_k := \varphi(x^k) - \varphi(x^*)$ *and* $u_k := t_k x^k - (t_k - 1)x^{k-1} - x^*$

Proof.

We apply Lemma 3 at the points $(x = x^k, y = y^{k+1})$ and $(x = x^*, y = y^{k+1})$ with $L = 1/\lambda^{k+1} = L^{k+1}$. We notice that $P_L(y^{k+1}) = prox_{\lambda g}(y^{k+1} - \lambda \nabla f(y^{k+1})) = x^{k+1}$. So we have:

$$\begin{aligned}
i) \quad & \frac{2}{L^{k+1}}(\varphi(x^k) - \varphi(x^{k+1})) \geq \|x^{k+1} - y^{k+1}\|^2 + 2\langle y^{k+1} - x^k, x^{k+1} - y^{k+1}\rangle \\
& \frac{2}{L^{k+1}}(v_k - v_{k+1}) \geq \|x^{k+1} - y^{k+1}\|^2 + 2\langle x^{k+1} - y^{k+1}, y^{k+1} - x^k\rangle \\
ii) \quad & \frac{2}{L^{k+1}}(-v_{k+1}) \geq \|x^{k+1} - y^{k+1}\|^2 + 2\langle x^{k+1} - y^{k+1}, y^{k+1} - x^*\rangle
\end{aligned}$$

We multiply (i) by $(t_{k+1} - 1)$ and add it to (ii):

$$\frac{2}{L^{k+1}}((t_{k+1}-1)v_k - t_{k+1}v_{k+1}) \geq (t_{k+1}-1)\|x^{k+1}-y^{k+1}\|^2 + 2(t_{k+1}-1) \\ \cdot \langle x^{k+1}-y^{k+1}, y^{k+1}-x^k\rangle + \|x^{k+1}-y^{k+1}\|^2 2\langle x^{k+1}-y^{k+1}, y^{k+1}-x^*\rangle \\ = t_{k+1}\|x^{k+1}-y^{k+1}\|^2 + 2\langle x^{k+1}-y^{k+1}, t_{k+1}y^{k+1}-(t_{k+1}-1)x^k-x^*\rangle \tag{2.6}$$

Now we notice that

$$t_{k+1}^2 - t_{k+1} = \left(\frac{1+\sqrt{1+4{t_k}^2}}{2}\right)^2 - \frac{1+\sqrt{1+4{t_k}^2}}{2} \\ = \frac{1}{4}(2+4t_k^2+2\sqrt{1+4t_k^2}-2-2\sqrt{1+4t_k^2}) = t_k^2$$

So we have

$$t_k^2 = t_{k+1}^2 - t_{k+1} \tag{2.7}$$

Multiplying 2.6 by t_{k+1} and using 2.7 we obtain:

$$\frac{2}{L^{k+1}}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|t_{k+1}(x^{k+1}-y^{k+1})\|^2 + \\ + 2t_{k+1}\langle x^{k+1}-y^{k+1}, t_{k+1}y^{k+1}-(t_{k+1}-1)x^k-x^*\rangle$$

Using the usual Pythagoras relation :

$$\|b-a\|^2 + 2\langle b-a, a-c\rangle = \|b-c\|^2 - \|a-c\|^2$$

with

$$a := t_{k+1}y^{k+1}, \qquad b := t_{k+1}x^{k+1}, \qquad c := (t_{k+1}-1)x^k + x^*$$

$$\frac{2}{L^{k+1}}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|t_{k+1}x^{k+1}-(t_{k+1}-1)x^k-x^*)\|^2 \\ - \|t_{k+1}y^{k+1}-(t_{k+1}-1)x^k-x^*\|^2$$

Now $t_{k+1}y^{k+1} = t_{k+1}x^k + (t_k-1)(x^k-x^{k+1})$ and we define $u_k := t_k x^k - (t_k-1)x^{k-1} - x^*$

Then

$$\begin{aligned}\frac{2}{L^{k+1}}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) &\geq \|u_{k+1}\|^2 - \|t_{k+1}x^k + (t_k-1)(x^k - x^{k-1}) - \\ &- (t_{k+1}-1)x^k - x^*\| \\ &= \|u_{k+1}\|^2 - \|t_k x^k - t_k x^{k-1} - x^k + x^{k-1} + x^k - x^*\|^2 \\ &= \|u_{k+1}\|^2 - \|t_k x^k - (t_k - 1)x^k - x^*\|^2 \\ &= \|u_{k+1}\|^2 - \|u_k\|^2\end{aligned}$$

Now $L^{k+1} \geq L^k$, so

$$\begin{aligned}\frac{2}{L^k}{t_k}^2 v_k - \frac{2}{L^{k+1}}{t_{k+1}}^2 v_{k+1} \geq \|u_{k+1}\|^2 - \|u_k\|^2 \\ 2\lambda^k {t_k}^2 v_k - 2\lambda^{k+1}{t_{k+1}}^2 v_{k+1} \geq \|u_{k+1}\|^2 - \|u_k\|^2\end{aligned}$$

□

Lemma 5. *let $\{a_k, b_k\}$ be positive sequences of real numbers satisfying $a_k - a_{k+1} \geq b_{k+1} - b_k\ \forall k \geq 1$, with $a_1 + b_1 \leq c$, $c > 0$, then $a_k \leq c\ \forall k \geq 1$.*

Proof. We prove by induction

- $a_1 \leq c - b_1 \leq c \implies a_1 \leq c$;

  $a_1 - a_2 \geq b_2 - b_1 \implies a_1 + b_1 - a_2 \geq b_2 \implies c \geq a_2 + b_2$
- we suppose true for $k-1$: $a_{k-1} + b_{k-1} \leq c$ and $a_{k-1} \leq c$

$$\begin{aligned}a_{k-1} - a_k \geq b_k - b_{k-1} \implies a_{k-1} + b_{k-1} - a_k \geq b_k \geq 0 \\ \implies a_k \leq a_{k-1} + b_{k-1} \leq c \implies a_k \leq c.\end{aligned}$$

□

Lemma 6. *the positive sequence $\{t_k\}_{k\geq 1}$ generated by FPGM with $t_1 = 1$ satisfies*

$$t_k \geq \frac{k+1}{2} \quad \forall k \geq 1 \tag{2.8}$$

Proof. We prove it by induction

- $t_1 = 1 = \frac{2}{2}$;
- we suppose true for k: $t_k \geq \frac{k+1}{2}$:

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2} \geq \frac{1+\sqrt{1+(k+1)^2}}{2} = \frac{1+\sqrt{k^2+2k+2}}{2} \geq \frac{1+k+1}{2}$$
$$= \frac{k+2}{2}$$

□

Theorem 3. *Let $\{x^k\}_{k\geq1}$ and $\{y^k\}_{k\geq1}$ be generated by FPGM. Then for any $k \geq 1$:*

$$\varphi(x^k) - \varphi(x^*) \leq \frac{2\alpha\mathcal{L}_f\|x^0 - x^*\|^2}{(k+1)^2}, \quad \forall k \geq 1.$$

Proof. Define

$$a_k := 2\lambda^k {t_k}^2 v_k, \quad b_k := \|u_k\|^2, \quad c := \|y^1 - x^*\|^2 = \|x^0 - x^*\|^2$$

Recall that $v_k = \varphi(x^k) - \varphi(x^*)$ an using lemma 4 we have $\forall k \geq 1$:

$$a_k - a_{k+1} \geq b_{k+1} - b_k$$

We note that $\{a_k, b_k\}$ are positive sequences of real and $c > 0$. Hence assuming that $a_1 + b_1 \leq c$. Invoking lemma 5 we have:

$$a_k \leq c \implies 2\lambda^k {t_k}^2 v_k \leq \|x^0 - x^*\|^2.$$

Using lemma 6:

$$v_k \leq \frac{2\|x^0 - x^*\|^2}{\lambda^k(k+1)^2}$$

We remember that $\lambda^k \in (0, 1/\mathcal{L}_f)$, so

$$\varphi(x^k) - \varphi(x^*) \leq \frac{2\alpha\mathcal{L}_f\|x^0 - x^*\|^2}{(k+1)^2}, \quad \forall k \geq 1.$$

All that remains is to prove the validity of the relation $a_1 + b_1 \leq c$. We can prove this applying lemma 3 with $x = x^*$, $y = y^1$ and $L = 1/\lambda^1$.

$$\begin{aligned}\varphi(x^*)-\varphi(x^1) &\geq \frac{L^1}{2}\|x^1-y^1\|^2+L\langle y^1-x^*,x^1-y^1\rangle\\ &=\frac{L^1}{2}\langle x^1-y^1,x^1-y^1\rangle+L\langle y^1-x^*,x^1-y^1\rangle\\ &=\frac{L^1}{2}\langle x^1-x^*+x^*-y^1,x^1-x^*+x^*-y^1\rangle+\\ &+L\langle y^1-x^*,x^1-x^*+x^*-y^1\rangle\\ &=\frac{L^1}{2}\|x^1-x^*\|^2-\frac{L^1}{2}\|y^1-x^*\|^2\\ &=\frac{L^1}{2}\big(\|x^1-x^*\|^2-\|y^1-x^*\|^2\big)\end{aligned}$$

Now

$$a_1=\frac{2}{L^1}v_1=\frac{2}{L^1}(\varphi(x^1)-\varphi(x^*)\leq\|y^1-x^*\|^2-\|x^1-x^*\|^2=c-b_1$$

So $a_1+b_1\leq c$.

□

2.3 Alternating direction method of multipliers (ADMM)

We consider the problem

$$minimize \quad \{f(x)+g(x)\}$$

where $f,g:\mathbb{R}^n\to\mathbb{R}\cup+\{\infty\}$ are closed proper convex functions and both f and g can be nonsmooth. The method is

$$\begin{aligned}x^{k+1}&:=prox_{\lambda f}(z^k-u^k)\\ z^{k+1}&:=prox_{\lambda g}(x^{k+1}+u^k)\\ u^{k+1}&:=u^k+x^{k+1}-z^{k+1}\end{aligned}$$

We notice that $x^k\in domf$ and $z^k\in domg$, so if g encodes constraints, the iterates z^k satisfy the constraints, while the iterates x^k satisfy the constraints only in the limit: for example if $g = \|\cdot\|_1$, then z^k will be sparse, while x^k will only be close to z^k, and so close to sparsity.

We observe that if f and g are the indicator functions of closed convex sets $\mathcal{C}$ and $\mathcal{D}$ respectively, the problem of minimizing $f + g$ is equivalent to the problem of finding a point $x \in \mathcal{C} \cap \mathcal{D}$. In fact, both proximal operators are reduced to projections.

2.3.1 Interpretations

The ADMM can be interpreted in many ways:

- **augmented Lagrangian**: we write the problem of minimizing $f(x) + g(x)$ as

$$\begin{array}{cl} minimize & f(x) + g(z) \\ s.t. & x - z = 0 \end{array} \tag{2.9}$$

which is called consensus form, in fact the variable x has been split into two variables x and z, and we have added the consensus constraint upon which they must agree. The augmented Lagrangian associated with the problem 2.9 is

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2 \tag{2.10}$$

where $\rho > 0$ is a parameter and $y \in \mathbb{R}^n$ is a dual variable associated with the consensus constraint. ADMM can then be expressed as

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

In each of the x and z steps, L_ρ is minimized over the variable, using the most recent value of the other primal variable and the dual variable. The dual variable is the scaled running sum of the consensus errors. Now we prove that the augmented Lagrangian form of ADMM reduces to the proximal version:

$$
\begin{aligned}
x^{k+1} &:= \underset{x}{\operatorname{argmin}}(f(x) + g(z^k) + {y^k}^T(x - z^k) + \frac{\rho}{2}\|x - z^k\|_2^2)) \\
z^{k+1} &:= \underset{z}{\operatorname{argmin}}(f(x^{k+1}) + g(z) + {y^k}^T(x^{k+1} - z) + \frac{\rho}{2}\|x^{k+1} - z\|_2^2)) \\
y^{k+1} &:= y^k + \rho(x^{k+1} - z^{k+1}).
\end{aligned}
$$

Then pull the linear terms into the quadratic ones and deleting the constant terms, we get

$$
\begin{aligned}
x^{k+1} &:= \underset{x}{\operatorname{argmin}}(f(x) + \frac{\rho}{2}\|x - z^k + \frac{1}{\rho}y^k\|_2^2)) \\
z^{k+1} &:= \underset{z}{\operatorname{argmin}}(g(z) + \frac{\rho}{2}\|x^{k+1} - z - \frac{1}{\rho}y^k\|_2^2)) \\
y^{k+1} &:= y^k + \rho(x^{k+1} - z^{k+1}).
\end{aligned}
$$

With $u^k = \frac{1}{\rho}y^k$ and $\lambda = \frac{1}{\rho}$, we obtain the proximal form of ADMM.

- **flow interpretation**: ADMM can also be interpreted as a method for solving a particular system of ordinary differential equations. Assuming for simplicity that f and g are both differentiable, we consider the differential equation

$$
\frac{d}{dt}\begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} -\nabla f(x(t)) - \rho u(t) - \rho r(t) \\ -\nabla g(z(t)) + \rho u(t) + \rho r(t) \end{bmatrix}, \quad \frac{d}{dt}u(t) = \rho r(t)
$$

where $r(t) = x(t) - z(t)$ is the primal residual and $\rho > 0$. With x^k, z^k and u^k denoting our approximations of $x(t)$, $z(t)$ and $u(t)$ at $t = kh$, where $h > 0$ is the step length, we use the discretization given by:

$$
\begin{aligned}
\frac{x^{k+1} - x^k}{h} &= -\nabla f(x^{k+1}) - \rho(x^k - z^k + u^k) \\
\frac{z^{k+1} - z^k}{h} &= -\nabla g(z^{k+1}) + \rho(x^{k+1} - z^k + u^k) \\
\frac{u^{k+1} - u^k}{h} &= \rho(x^{k+1} - z^{k+1})
\end{aligned}
$$

We make very specific choices on the righthand side as to whether each time argument t is replaced with kh (forward) or $(k+1)h$ (backward) values. Choosing $h = \lambda$ and $\rho = 1/\lambda$, this discretization reduces directly to the proximal form of ADMM.

- **fixed point iteration**: ADMM can be viewed as a fixed point iteration for finding a point x^* satisfying the optimality condition

$$0 \in \partial f(x^*) + \partial g(x^*) \tag{2.11}$$

Fixed points x, z, y of the ADMM iteration satisfy:

$$x = proz_{\lambda f}(z - u), \quad z = prox_{\lambda,g}(x + u), \quad u = u + x - z$$

from the last equation, we conclude $x = z$, so

$$x = proz_{\lambda f}(x - u), \quad x = prox_{\lambda,g}(x + u)$$

which can be written as

$$x = (I + \lambda\partial f)^{-1}(x - u), \quad x = (I + \lambda\partial g)^{-1}(x + u)$$

This is the same as

$$x - u \in x + \lambda\partial f(x), \quad x + u \in x + \lambda\partial g(x)$$

Adding these two equations we have:

$$0 \in \lambda\partial f(x) + \lambda\partial g(x)$$

so x satisfies the optimality condition 2.11. Thus any fixed point of the ADMM iteration satisfies $x = z$ with x optimal.

2.3.2 Linearized ADMM

A variation of ADMM can be useful for solving problems of the form

$$minimize f(x) + g(Ax)$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and $g : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ are closed proper convex and $A \in \mathbb{R}^{m\times n}$. This problem can be solved with the standard ADMM by defining $\tilde{g}(x) := g(Ax)$ and minimizing $f(x) + \tilde{g}(x)$. However, this approach requires evaluation of the proximal operator of $\tilde{g}$ and it can be complicated by the presence of A. The linearized ADMM algorithm solves the problem above using only the proximal operator of f, g and the multiplication by A and A^T.

Linearized ADMM has the form:

$$\begin{aligned} x^{k+1} &:= prox_{\mu f}\left(x^k - \frac{\mu}{\lambda} A^T(Ax^k - z^k + u^k)\right) \\ z^{k+1} &:= prox_{\lambda g}(Ax^{k+1} + u^k) \\ u^{k+1} &:= u^k + Ax^{k+1} - z^{k+1} \end{aligned}$$

where the parameters λ and μ satisfy $0 < \mu \leq \lambda \|A\|_2^2$. This reduces to standard ADMM when $A = I$ and $\mu = \lambda$.

The reason for the name is the following: consider the problem

$$\begin{array}{cl} minimize & f(x) + g(z) \\ s.t. & Ax - z = 0 \end{array}$$

with variables x and z. The augmented Lagrangian for this problem is

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax - z) + \frac{\rho}{2}\|Ax - z\|_2^2$$

where $y \in \mathbb{R}^m$ is the dual variable and $\rho = 1/\lambda$. In the linearized ADMM we modify the x-update by replacing the term $(\rho/2)\|Ax - z^k\|_2^2$ with

$$\rho(A^T Ax^k - A^T z^k)^T x + \frac{\mu}{2}\|x - x^k\|_2^2$$

i.e. we linearize the quadatic term and add new quadratic regularization. So, the augmented Lagrangian becomes

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax - z) + \rho(A^T Ax^k - A^T z^k)^T x + \frac{\mu}{2}\|x - x^k\|_2^2$$

$$\begin{aligned}
x^{k+1} &= \operatorname*{argmin}_x L_\rho(x, z^k, y^k) \\
&= \operatorname*{argmin}_x f(x) + {y^k}^T Ax + \rho(A^T Ax^k - A^T z^k)^T x + \frac{\mu}{2}\|x - x^k\|_2^2 \\
&= \operatorname*{argmin}_x f(x) + \frac{1}{2\mu}\|x - x^k + \rho\mu A^T(Ax^k - z^k + \frac{y^k}{\rho})\|_2^2 \\
&= prox_{\mu f}\left(x^k - \frac{\mu}{\lambda}A^T(Ax^k - z^k + u^k)\right)
\end{aligned}$$

with $u^k = (1/\rho)y^k$ and $\lambda = 1/\rho$.

2.3.3 Convergence

There are many convergence results about the ADMM algorithm discussed in the literature. We will make two assumptions:

- the functions f and g are closed, proper and convex;
- the Lagrangian L_0 2.10 has a saddle point.

The first assumption implies that the subproblems arising in the x-update and z-update are solvable, i.e. there exist x and z, not necessarily unique, which minimize the augmented Lagrangian. From the second assmption we have that there exists (x^*, z^*, y^*) saddle point, for which

$$L_0(x^*, z^*, y) \leq L_0(x^*, z^*, y^*) \leq L_0(x, z, y^*), \quad \forall x, z, y.$$

By assumption 1, it follows that $L_0(x^*, z^*, y^*)$ is finite for any saddle point, and it implies that (x^*, z^*) is a solution to 2.9, so $x^* - z^* = 0$, $f(x^*) < \infty$ and $g(z^*) < \infty$. From assumption 2 we have

$$\max_y L_0(x^*, z^*, y) = \min_{x,z} L_0(x, z, y^*) = L_0(x^*, z^*, y^*)$$

so it implies that y^* is dual optimal solution. For the strong duality theorem we also have that the optimal values of the primal and dual problems are equal. Under these assumptions, the ADMM iterates satisfy the following:

- residual convergence : $r^k = x^k - z^k \to 0$ as $k \to +\infty$;

- objective convergence: let $p^* = min\{f(x) + g(z) : x - z = 0\}$, then $f(x^k) + g(z^k) \to p^*$ as $k \to +\infty$, i.e. the objective function of the iterates approaches the optimal value;
- dual variable: $y^k \to y^*$ as $k \to +\infty$, where y^* is a dual optimal solution.

2.4 Proximal Newton method

In this section we consider a line search method to solve the problem

$$\min\{\varphi(x) = f(x) + g(x)\},$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a proper convex, continuosly differentiable function and its gradient is Lipshitz continuos; instead $g : \mathbb{R}^n \to \mathbb{R}$ is a proper convex, but not necessarly differentiable function, whose proximal mapping can be evaluated efficiently. Here g is a penalty function. Asssuming that the optimal value f^* is attained at some optimal solution x^*, not necessarly unique.

We remember that a line search algorithm follows a iterative pattern, where in each iteration k:

- we choose a descent direction d^k, where d is a descent direction for φ in x if we find $\bar{t} > 0$ such that $\varphi(x + td) < \varphi(x)$ $\forall t \in (0, \bar{t}]$;
- we find a step length t^k ;
- we update $x^{k+1} = x^k + t^k d^k$.

Now we recall the motivation for the proximal gradient algorithm: we iteratively minimize a quadratic expansion of f plus original g

$$\begin{aligned} x^{k+1} &= \underset{z}{\operatorname{argmin}}\big\{\nabla f(x^k)^T(z - x^k) + \frac{1}{2\lambda}\|z - x^k\|^2 + g(x^k)\big\} \\ &= prox_{\lambda g}(x^k - \lambda\nabla f(x^k)) \end{aligned}$$

A fundamental difference between PGM and Newton's method is that the latter uses the local hessian of f, instead the PGM uses hessian equal to the identity $\frac{1}{\lambda}I$. So what happens if we substitute $\frac{1}{\lambda}I$ with$\nabla^2 f$?

This leads us to the proximal Newton method (PNM). Starting with x^0, we repeat for $k \geq 1$:

$$d^k = \operatorname*{argmin}_d \big\{ \nabla f(x^{k-1})^T d + \frac{1}{2} d^T H^{k-1} d + g(x^{k-1} + d) \big\}$$
$$x^k = x^{k-1} + t^k d^k$$

Here H^k is $\nabla^2 f(x^k)$ and we have the proximal Newton method, or it is an approximation to $\nabla^2 f(x^k)$ and we have the proximal quasi-Newton method.

An equivalent formulation is :

$$\Delta^k = \operatorname*{argmin}_z \big\{ \nabla f(x^{k-1})^T (z - x^{k-1}) + \frac{1}{2} (z - x^{k-1})^T H^{k-1} (z - x^{k-1}) + g(z) \big\} - x^{k-1}$$
$$x^k = x^{k-1} + t^k \Delta^k$$

Now we must define the scaled proximal mapping

$$prox_g^H(x) := \operatorname*{argmin}_z \big\{ g(z) + \frac{1}{2} \|x - z\|_H^2 \big\}$$

where $\|x\|_H^2 = x^T H x$ defines a norm given a matrix $H \succ 0$, i.e. H is a positive definite matrix (all of the eigenvalues of H are positive).

We note that it exists an unique $z \in dom(g)$ for all $x \in dom(f)$, because proximity function is strongly convex if $H \succ 0$.

If we denote $prox_g^H(x)$ with z we have

$$0 \in \partial g(z) + H(z - x) \iff H(x - z) \in \partial g(z) \iff H(x - prox_g^H(x)) \in \partial(prox_g^H(x)) \tag{2.12}$$

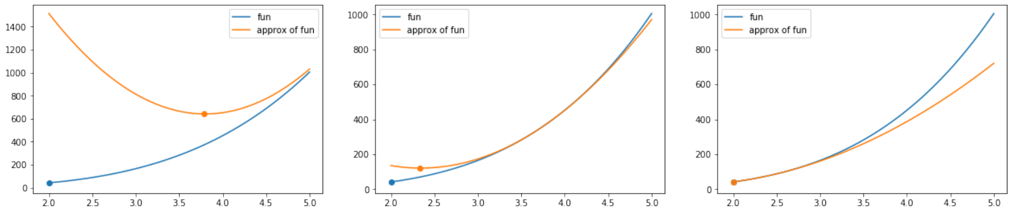
We note that:

$$\begin{aligned}
&prox_g^H(x - H^{-1}\nabla f(x)) = \operatorname*{argmin}_y \big\{ g(y) + \frac{1}{2} \|y - x + H^{-1}\nabla f(x)\|_H^2 \big\} \\
&= \operatorname*{argmin}_y \big\{ g(y) + \frac{1}{2} (y - x + H^{-1}\nabla f(x))^T H (y - x + H^{-1}\nabla f(x)) \big\} \\
&= \operatorname*{argmin}_y \big\{ g(y) + \frac{1}{2} (y - x)^T H (y - x) + \frac{1}{2} (y - x)^T H H^{-1} \nabla f(x) \\
&+ \frac{1}{2} \nabla f(x)^T H^{-T} H (y - x) + \frac{1}{2} \nabla f(x)^T H^{-T} H H^{-1} \nabla f(x) \big\} \\
&= \operatorname*{argmin}_y \{ \nabla f(x)^T (y - x) + \frac{1}{2} (y - x)^T H (y - x) + g(y) \}
\end{aligned}$$

So we can rewrite:

$$\Delta x^k = prox_g^H(x^{k-1} - H^{-1}\nabla f(x^{k-1})) - x^{k-1}$$
$$x^k = x^{k-1} + t^k \Delta x^k$$

We make an easy example: let $f(x) = x^4 + 3x^3$ and $g(x) = |x|$. Our problem is $\min f(x) + g(x)$. We consider only the interval $[2, 5]$, in this way f is convex. The minimum is $x = 2$. We apply the algorithm starting with $x^0 = 6$ and we obtain $x^1 = 3.79$, $x^2 = 2.33$, $x^3 = 2$.



Now $\Delta x = prox_g^H(x - H^{-1}\nabla f(x)) - x$ is the proximal Newton-type search direction. Using property 2.12 we see that

$$H(x - H^{-1}\nabla f(x) - \Delta x - x) \in \partial g(\Delta x + x) \implies H(-H^{-1}\nabla f(x) - \Delta x) \in \partial g(\Delta x + x)$$

$$-\nabla f(x) - H\Delta x \in \partial g(\Delta x + x) \implies H\Delta x \in -\nabla f(x) - \partial g(\Delta x + x)$$

We notice that it combines an explicit gradient with an implicit subgradient. In fact, using the scaled proximal mapping we can interpret the search direction as the composition of Newton's step with a proximal operator.

If $\boldsymbol{g \equiv 0}$, we have $H\Delta x \in -\nabla f(x) \implies \Delta x \in -H^{-1}\nabla f(x)$, so the PNM is the classical Newton system.

Now we prove that Δx is a descent direction:

Lemma 7. *if $H \succ 0$, then*

$$\varphi(x + t\Delta x) \leq \varphi(x) + t(\nabla f(x)^T \Delta x + g(x + \Delta x) - g(x)) + O(t^2) \tag{2.13}$$

$$\nabla f(x)^T \Delta x + g(x + \Delta x) - g(x) \leq -\Delta x^T H \Delta x \tag{2.14}$$

Proof. Assuming $t \in (0,1]$.

$$\begin{aligned}\varphi(x+t\Delta x)-\varphi(x) &= f(x+t\Delta x)-f(x)+g(x+t\Delta x)-g(x)\\ &= f(x+t\Delta x)-f(x)+g(x+t\Delta x-tx+tx)-g(x)\\ &= f(x+t\Delta x)-f(x)+g((1-t)x+t(\Delta x+x))-g(x)\\ &\leq f(x+t\Delta x)-f(x)+tg(\Delta x+x)-tg(x)\\ &= \nabla f(x)^T(t\Delta x)+tg(\Delta x+x)-tg(x)+O(t^2)\\ &= t(\nabla f(x)^T\Delta x+g(x+\Delta x)-g(x))+O(t^2)\end{aligned}$$

Since Δx is minimum :

$$\begin{aligned}&\nabla f(x)^T\Delta x+\frac{1}{2}\Delta x^T H\Delta x+g(x+\Delta x)\\ &\leq \nabla f(x)^T t\Delta x+\frac{1}{2}t^2\Delta x^T H\Delta x+g(x+t\Delta x)\\ &\leq t\nabla f(x)^T\Delta x++\frac{1}{2}t^2\Delta x^T H\Delta x+tg(\Delta x+x)+(1-t)g(x)\end{aligned}$$

Then

$$(1-t)\nabla f(x)^T\Delta x+\frac{1}{2}(1-t^2)\Delta x^T H\Delta x+(1-t)(g(x+\Delta x)-g(x))\leq 0$$

$$\nabla f(x)^T\Delta x+\frac{1}{2}(1+t)\Delta x^T H\Delta x+g(x+\Delta x)-g(x)\leq 0$$

$$\nabla f(x)^T\Delta x+g(x+\Delta x)-g(x)\leq -\frac{1}{2}(1+t)\Delta x^T H\Delta x \xrightarrow[t\to 1]{} -\Delta x^T H\Delta x$$

□

If we replace 2.14 in 2.13 we have $\varphi(x+t\Delta x)\leq\varphi(x)-t\Delta x^T H\Delta x+O(t^2)<\varphi(x)$ because H is positive definite, so $\Delta x^T H\Delta x\geq 0$. In this way, Δx is a descent direction.

Now we talk about the method to find the step length t^k, we use a backtracking line search. It is an algorithm to determine the maximum amount to move along a given search direction. It chooses the t such that

$$\varphi(x+t\Delta x)\leq\varphi(x)+\alpha t\lambda, \tag{2.15}$$

where $\lambda=\nabla f(x)^T\Delta x+g(x+\Delta x)-g(x)$ and $\alpha\in(0,1/2)$ is a control parameter. From a practical point of view we start wih $t=1$ and if the descent condition is not satisfied we halve the step.

We can prove a condition about the step length:

Lemma 8. *Suppose $H \succ mI$ for some $m > 0$, ∇f Lipschitz with constant L, then $t \leq \min\{1, \frac{2m}{L}(1-\alpha)\}$ satisfies the sufficient descent condition 2.15.*

Proof.

$$\begin{aligned}
\varphi(x+t\Delta x)-\varphi(x) &= f(x+t\Delta x)-f(x)+g(x+t\Delta x)-g(x)\\
&\leq \int_0^1 \nabla f(x+s(t\Delta x))^T t\Delta x ds + tg(x+\Delta x)+(1-t)g(x)-g(x)\\
&= \nabla f(x)^T(t\Delta x)+t[g(x+\Delta x)-g(x)]+\int_0^1(\nabla f(x+s(t\Delta x))-\\
&\qquad \nabla f(x))^T t\Delta x ds\\
&\leq t(\nabla f(x)^T\Delta x+g(x+\Delta x)-g(x)+\int_0^1 \|\nabla f(x+s(t\Delta x))-\\
&\qquad \nabla f(x)\|\|\Delta x\|ds)\\
&\leq t(\nabla f(x)^T\Delta x+g(x+\Delta x)-g(x)+\frac{Lt}{2}\|\Delta x\|^2)\\
&t(\lambda+\frac{Lt}{2}\|\Delta x\|^2)
\end{aligned}$$

If we choose $t \leq \frac{2m}{L}(1-\alpha)$

$$\frac{Lt}{2}\|\Delta x\|^2 \leq m(1-\alpha)\|\Delta x\|^2 \leq (1-\alpha)\Delta x^T H\Delta x \leq -(1-\alpha)\lambda$$

So $\varphi(x+t\Delta x)-\varphi(x) \leq t(\lambda-(1-\alpha)\lambda) = t\alpha\lambda$. □

Then, in conclusion, the algorithm of PNM is:

```
given $x^0 \in dom(\varphi)$
repeat until stopping conditions are satisfied
    1. let $H^k$ approximation of hessian $\nabla^2 f(x^k)$
    2. solve the subproblem $d^k = \text{argmin}_d \nabla f(x^k)^T d + \frac{1}{2}d^T H^k d + g(x^k+d)$
    3. select $t^k$ with backtracking line search
    4. update $x^{k+1} = x^k + t^k d^k$
```

2.4.1 Proximal quasi-Newton method

The proximal quasi-Newton method is a quasi-Newton method where the search direction is found using a proximal operator. They are algorithms where the Hessian matrix does not need to be computed, but it is updated by analyzing successive gradient vectors instead.

Let H^k an approximation of the Hessian of $f(x^k)$, x^k and x^{k+1} points at k and $k+1$ -th iterates and h^k, h^{k+1} the gradients at k and $k+$ -th iterates ($h^k = \nabla f(x^k)$). Now, the quasi-Newton methods represent a generalization of the secant method to find the root of the first derivative for multidimensional problem, so the matrix H^{k+1} must satisfy the condition

$$H^{k+1}(x^{k+1} - x^k) = h^{k+1} - h^k \tag{2.16}$$

C. G. Broyden suggested to use, as update of the Hessian, the current estimate of the matrix H^k and improving upon it by taking the solution to the secant equation:

$$H^{k+1} = H^k + \frac{(y_k - H^k s_k)s_k^T}{s_k^T s_k}$$

where

$$y_k := h^{k+1} - h^k = \nabla f(x^{k+1}) - \nabla f(x^k), \quad s_k := x^{k+1} - x^k$$

Now in the formula of the proximal Newton method, the matrices H^k should be invertible. Hence we remember the Sherman- Morrison formula:

Lemma 9. *For $u, v \in \mathbb{R}^n$ the matrix $(I + uv^T)$ is invertible if and only if $1 + u^T v \neq 0$ and in that case*

$$(I + uv^T)^{-1} = I - \frac{1}{1 + u^T v} uv^T$$

If the matrix H^k is nonsingular, then for the previous lemma the matrix

$$H^{k+1} = H^k \left(I + \frac{(H^k)^{-1}(y_k - H^k s_k)s_k^T}{\|s_k\|_2^2} \right)$$

is non-singular if and only if

$$1 + \frac{((H^k)^{-1}(y_k - H^k s_k))^T s_k}{\|s_k\|_2^2} \neq 0 \quad \Longrightarrow \quad \|s_k\|_2^2 + ((H^k)^{-1}(y_k - H^k s_k))^T s_k \neq 0$$

In which case, using the lemma, the inverse is

$$(H^{k+1})^{-1} = \left[I - \frac{((H^k)^{-1}(y_k - H^k s_k))s_k^T}{\|s_k\|_2^2 + ((H^k)^{-1}(y_k - H^k s_k))^T s_k}\right](H^k)^{-1}$$

We denote with $B^k := (H^k)^{-1}$ and $B^{k+1} := (H^{k+1})^{-1}$, then this can be written in the form

$$B^{k+1} = B^k + \frac{(s_k - B^k y^k)s_k^T}{s_k^T B^k y^k} B^k$$

Now if the matrix B^k is symmetric and definite positive we are not sure that B^{k+1} is symmetric or definite positive. We want these properties because the matrices $\{B^k\}_k$ are the approximations of the inverse of the Hessian of a convex function. To recover these properties we update the form

$$B^{k+1} = B^k + \alpha u u^T + \beta v v^T$$

Imposing the secant condition 2.16 on the inverse

$$\begin{aligned} B^{k+1} y^k = s_k &\implies B^k y^k + \alpha(u^T y^k)u + \beta(v^T y^k)v = s_k \\ &\implies \alpha(u^T y^k)u + \beta(v^T y^k)v = s_k - B^k y^k \end{aligned}$$

This equation has not a unique solution. A choice can be $u = s_k$ and $v = B^k s_K$:

$$\alpha(s_k^T y^k)s_k + \beta(y^k B^k s_K)B^k s_K = s_k - B^k y^k$$

We obtain

$$\alpha = \frac{1}{s_k^T y^k} \quad \beta = -\frac{1}{(y^k)^T B^k y^k}$$

Substituting in the updated formula, we obtain the Davidon-Fletcher-Powell (DFP) formula:

$$B^{k+1} = B^k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{B^k y^k (y^k)^T B^k}{(y^k)^T B^k y^k}$$

This is only one of the possible choices: with other solutions we obtain different formulas.

Theorem 4. *Given B^k symmetric and positive definite, then the DFP update produces B^{k+1} positive definite if and only if $s_k^T y^k > 0$.*

Another update which maintain symmetry and positive definitiveness is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula. It was independently discovered by the four authors. A way to introduce this formula is by the concept of duality. We consider an update

$$H^{k+1} = \mathcal{U}(H^k, s_k, y^k)$$

which satisfies $H^{k+1}s_k = y^k$ 2.16. Then by exchanging $H^k \Leftrightarrow B^k$ and $s_k \Leftrightarrow y^k$ we obtain the dual update for the inverse of the Hessian, i.e.

$$B^{k+1} = \mathcal{U}(B^k, y^k, s_k)$$

which satisfies $B^{k+1}y^k = s_k$. If we start from the DFP updated formula, by duality we obtain the BFGS updated formula

$$H^{k+1} = H^k + \frac{y^k (y^k)^T}{(y^k)^T s^k} - \frac{H^k s_k s_k^T H^k}{s_k^T H^k s_k}$$

The BFGS formula written in this way is not useful in the case of large problems. We need an equivalent formulation; this can be done using a generalization of the Sherman-Morrison formula: (Sherman-Morrison-Woodbury)

Lemma 10.

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}UC^{-1}V^T A^{-1} \quad \textit{where} \quad C = I + V^T A^{-1} U$$

with $U = [u_1, u_2, ..., u_k]$ and $V = [v_1, v_2, ..., v_k]$.

So, by using the Sherman-Morrison formula, the BFGS update becomes

$$B^{k+1} = \left(I - \frac{s_k (y^k)^T}{s_k^T y^k}\right) B^k \left(I - \frac{y^k s_k^T}{s_k^T y^k}\right) + \frac{s_k s_k^T}{s_k^T y^k}$$

Proof. Let $U = [u_1, u_2]$ and $V = [v_1, v_2]$ with

$$u_1 = v_1 = \frac{y^k}{(s_k^T y^k)^{1/2}}, \quad u_2 = -v_2 = \frac{H^k s_k}{(s_k^T H^k s_k)^{1/2}}$$

We compute the matrix C of the Sherman-Morrison formula:

- $C_{11} = 1 + v_1^T (H^k)^{-1} u_1 = 1 + \frac{(y^k)^T B^k y^k}{s_k^T y^k} = \beta$
- $C_{22} = 1 + v_2^T (H^k)^{-1} u_2 = 0$
- $C_{12} = \frac{(s_k^T y^k)^{1/2}}{(s_k^T H^k s_k)^{1/2}}$
- $C_{21} = -C_{12} = -\alpha$

Then

$$C = \begin{bmatrix} \beta & \alpha \\ -\alpha & 0 \end{bmatrix}, \quad C^{-1} = \frac{1}{\alpha^2} \begin{bmatrix} 0 & -\alpha \\ \alpha & \beta \end{bmatrix}$$

Let $\tilde{U} = B^k U$ and $\tilde{V} = B^k V$, using Sherman-Morris formula

$$\begin{aligned} B^{k+1} &= B^k - B^k U C^{-1} V^T B^k = B^k - \tilde{U} C^{-1} \tilde{V}^T \\ \tilde{U} C^{-1} \tilde{V}^T &= \frac{1}{\alpha^2} [\tilde{u}_1, \tilde{u}_2] \begin{bmatrix} 0 & -\alpha \\ \alpha & \beta \end{bmatrix} \begin{bmatrix} \tilde{v}_1^T \\ \tilde{v}_2^T \end{bmatrix} = \\ &= \frac{1}{\alpha} (B^k u_2 v_1^T B^k - B^k u_1 v_2^T B^k) + \frac{\beta}{\alpha} (B^k u_2 v_2^T B^k) \end{aligned}$$

If we substitute the values of α, β, u_1, u_2, v_1 and v_2, we obtain

$$B^{k+1} = B^k - \frac{B^k y^k s_k^T + s_k (y^k)^T B^k}{s_k^T y^k} + \frac{s_k s_k^T}{s_k^T y^k} \left(1 + \frac{(y^k)^T B^k y^k}{s_k^T y^k} \right)$$

or equivalently

$$B^{k+1} = \left(I - \frac{s_k (y^k)^T}{s_k^T y^k} \right) B^k \left(I - \frac{y^k s_k^T}{s_k^T y^k} \right) + \frac{s_k s_k^T}{s_k^t y^k}$$

□

Theorem 5. *Given B^k symmetric and positive definite, then the BFGS update produces B^{k+1} positive definite id and only id $s_k^T y^k > 0$.*

We can describe both the update (DFP and BFGS) in an unique way: let B_{DFP}^{k+1} the DFP update and B_{BFGS}^{k+1} the BFGS update, then the following update

$$B_\theta^{k+1} = (1-\theta)B_{DFP}^{k+1} + \theta B_{BFGS}^{k+1}$$

maintains for any θ the symmetry and for any $\theta \in [0,1]$ the positive definitiveness. For the provious theorem, the update is positive definite for any $\theta \in [0,1]$ if and only if $s_k^T y^k > 0$. Equivalently

$$H_\theta^{k+1} = (1-\theta)H_{DFP}^{k+1} + \theta H_{BFGS}^{k+1}$$

Then, the proximal quasi-Newton algorithm is:

```
given $x^0 \in dom(\varphi)$
repeat until stopping conditions are satisfied
    1. let $H^k$ computed using DFP or BFGS
    2. solve the subproblem $d^k = \text{argmin}_d \nabla f(x^k)^T d + \frac{1}{2} d^T H^k d + g(x^k + d)$
    3. select $t^k$ with backtracking line search
    4. update $x^{k+1} = x^k + t^k d^k$
```

2.4.2 Inexact proximal Newton method

In the algorithm in practice it is expensive to compute the solution accurately, because we must solve the subproblem (2.) $\text{argmin}_z \{\nabla f(x^{k-1})^T(z - x^{k-1}) + \frac{1}{2}(z - x^{k-1})^T H^{k-1}(z - x^{k-1}) + g(z)\}$. In order to make this approach efficient in practice, we perform this inner minimization inexactly. We talk about inexact proximal Newton method. We can use, for example the PGM or the FPGM to find this inexact descent direction. However, to compute an adequate approximate solution we need some measure of closeness to optimality.

We recall that an iteration of PGM is

$$x^{k+1} = prox_{\lambda^k g}(x^k - \lambda^k \nabla f(x^k))$$

An equivalent formulation is

$$\begin{aligned} x^{k+1} &= x^k - \lambda^k G_{\lambda,\varphi}(x^k) \\ G_{\lambda,\varphi}(x^k) &= \frac{1}{\lambda^k}(x^k - prox_{\lambda^k g}(x^k - \lambda^k \nabla f(x^k))) \end{aligned}$$

where $G_\varphi(x^k)$ is a generalized gradient step, in fact $G_\varphi(x) = 0$ if and only if x minimizes φ, so $\|G_\varphi(x)\| = 0$ generalizes the smooth first order measure of optimality $\|\nabla\varphi(x)\|$.

An early stopping condition for the subproblem is based on two ideas:

- the approximation of f must be accurate
- near a solution, the subproblem should be solved almost exactly

We thus require that the solution $\tilde{z^k}$ of the k-th subproblem satisfy

$$\|G_{\tilde{\varphi}_k/M}(\tilde{z^k})\| \leq \eta_k \|G_{\varphi/M}(x^k)\| \tag{2.17}$$

where $\tilde{\varphi}_k(x) = f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T H^k (x - x^k) + g(x) =: F_k(x) + g(x)$ is the approximation of φ, η_k is a forcing term and $mI \preceq H^k \preceq MI$. We choose the forcing term based of the agreement between f and the previsious quadratic approximation F_{k-1}. We set $\eta_1 := 0.5$ and

$$\eta_k := min\{\frac{m}{2}, \frac{\|G_{\varphi_{\tilde{k-1}}/M}(x^k) - G_{\varphi/M}(x^k)\|}{\|G_{\varphi/M}(x^{k-1})\|}\}$$

This choice due to Eisenstat and Walker in "Choosing the forcing terms in an exact Newton method ", yields desiderable convergence results and performs admirably in practice.

However, depending on the method used to solve the subproblem, we can not have a descent direction. For example, in the previous chapter we said that the FPGM is not a descent algorithm, in fact some ripples can be formed. To avoid this, we impose the additional condition that the quadratic model is decreased, i.e., if $\tilde{z}$ is the solution of the subproblem we must have that $\varphi_k(\tilde{z}) \leq \varphi_k(x^k)$, in this way $\tilde{z} - x^k$ is a descent direction.

With these conditions the inexact proximal Newton method is similar to a trust-region algorithm, in fact at every iteration we search a direction which is inside the region of the descent directions such that $\|G_{\tilde{\varphi}_k}(\cdot)\| \leq \eta_k \|G_\varphi(\cdot)\|$. We can adjust the trust-region size using the forcing term, which is the ratio between the reduction predicted from the model and the true reduction.

2.4.3 Convergence

We assume that the Hessian approximations are sufficiently positive definite, i.e. $H^k \succeq mI$ $k = 1, 2, ...$ for some $m > 0$; then from Lemma 8 there exists a step length that satisfies the decrease condition 2.15. Therefore, x is a minimizer of $f + g$ if and only if the search direction is zero, i.e. $0 = \operatorname{argmin}_d \nabla f(x)^T d + \frac{1}{2} d^T H d + g(x+d)$. In this way the global convergence of proximal Newton method results from the fact that the search direction is a descent direction and if H^k are sufficiently positive definite then the step lengths are bounded away from zero

Theorem 6. *Suppose $H^k \succeq mI$ $k = 1, 2, ...$ for some $m > 0$. Then the sequence $\{x^k\}$ generated by a proximal Newton method converges to a minimizer of $\varphi = f + g$.*

Proof. The sequence $\{\varphi(x^k)\}$ is decreasing because Δx^k are descent directions and there exist step lengths satisfying descent condition 2.15:

$$\varphi(x^{k+1}) - \varphi(x^k) \leq \alpha t^k \lambda^k \leq 0$$

The sequence $\{\varphi(x^k)\}$ must converge to some limit because f is closed and the optimal value is attained. Thus $t^k \lambda^k$ must go to zero. The step lengths t^k are bounded away from zero from Lemma 8, therefore λ^k must decay to zero. Now

$$\|\Delta x^k\|^2 \leq \frac{1}{m} \Delta x^{k^T} H^k \Delta x^k \leq -\frac{1}{m} \lambda^k$$

thus Δx^k also converges to zero. Since the search direction is zero if and only if x is an optimal solution, x^k must converge to some optimal solution x^*.

□

Now we talk about the convergence rate of the proximal Newton methods when the subproblems are solved exactly. First, we state our assumptions on the problem:

- f is twice-continuosly differentiable;
- f is strongly convex with constant m, i.e. $f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|x - y\|^2$ for any x, y;
- ∇f and $\nabla^2 f$ are Lipschitz continuous with constants L_1 ans L_2.

Under the above assumptions we have that the **proximal Newton method** converges quadratically to x^*, i.e.

$$\|x^{k+1} - x^*\| \leq \frac{L_2}{2m}\|x^k - x^*\|^2.$$

Instead there is a different local convergence for **proximal quasi-Newton method**: under the same assumptions, if the sequence $\{H^k\}$ satisfies the Dennis-Moré criterion, namely

$$\frac{\|(H^k - \nabla^2 f(x^*))(x^{k+1} - x^k)\|}{\|x^{k+1} - x^k\|} \to 0$$

and if $mI \preceq H^k \preceq MI$ for some $0 < m \leq M$, then a proximal quasi-Newton method converges superlinearly to x^*, i.e.

$$\|x^{k+1} - x^*\| \leq o(\|x^k - x^*\|).$$

Otherwise the **inexact proximal Newton method** with unit step length under the same assumptions and if the starting point x^0 is close to x^*:

- if η_k is smaller than some $\bar{\eta} < \frac{m}{2}$, it converges quadratically to x^*;
- if η_k decays to zero, it converges superlinearly to x^*.

2.4.4 Self-concordant functions

If proximal Newton method is applied for minimizing a sum of a quadratic function and a convex function with an inexpensive proximal operator, we can prove that the convergence is very fast. In fact we suppose that f is a quadratic function, i.e. $f(x) = \frac{1}{2}x^T Ax + b^T x + c$, where A is a $n \times n$ definite positive real matrix, $b \in \mathbb{R}^n$ is a vector and $c \in \mathbb{R}$. Let g a convex function, it can be nonsmooth. We have

$$\nabla f(x) = Ax + b \quad \nabla^2 f(x) = A$$

Since f is quadratic, if we apply an iteration of the standard Newton's method with $g \equiv 0$ we find the optimal value: given x^0 a starting point

$$x^1 = x^0 - (\nabla^2 f(x^0))^{-1}\nabla f(x^0) = x^0 - A^{-1}(Ax^0 + b)) = x^0 - x^0 - A^{-1}b = -A^{-1}b$$

In the case $g \neq 0$, the descent direction of the proximal Newton method becomes

$$\Delta x^0 = prox_g^{\nabla^2 f}(x^0 - (\nabla^2 f(x^0))^{-1}\nabla f(x^0)) - x^0 = prox_g^A(-A^{-1}b) - x^0$$

Then

$$x^1 = x^0 + t^0(prox_g^A(-A^{-1}b) - x^0)$$

The algorithm is very easy, if the proximl operator of g is easy to compute.

Therefore if we have a quadratic function the PNM converge rapidly. We note that the algorithm is characterized by the fact that the hessian of the function f is the same at each iterations, so we want to enlarge this idea. By extension, if the hessian matrix does not change quickly, then PNM converges fast. Thus, the algorithm performs "well" if small changes in x lead to small changes in the second derivative. Change in second derivative can be measured using the third derivative, so third derivative should be small relative to the second derivative. The self-concordant function reflects this requirement.

Definition 7. *A function $f : \mathbb{R} \to \mathbb{R}$ is self-concordant if*

- *f is convex and three times derivable*
- *$|f^{'''}(x)| \leq 2f^{''}(x)^{3/2}$ for all $x \in dom(f)$.*

The constant 2 in the definition can be replaced with another constant k, we choose $k = 2$ because in this way the function $f(x) = -log(x)$ for $x > 0$ is self-concordant without any scaling. We make same example:

- linear and quadratic function are self concordant, in fact $f^{'''}(x) = 0$ for all x;
- negative logarithm $f(x) = -log(x)$, $x > 0$ is self-concordant:

$$f^{''}(x) = \frac{1}{x^2}, \quad f^{'''}(x) = -\frac{2}{x^3}, \quad \frac{|f^{'''}(x)|}{f^{''}(x)^{3/2}} = 2;$$

- exponential function e^x is not self-concordant: $f^{''}(x) = f^{'''}(x) = e^x$

$$\frac{|f^{'''}(x)|}{f^{''}(x)^{3/2}} = \frac{e^x}{e^{3x/2}} = e^{-x/2} \xrightarrow[x \to -\infty]{} \infty.$$

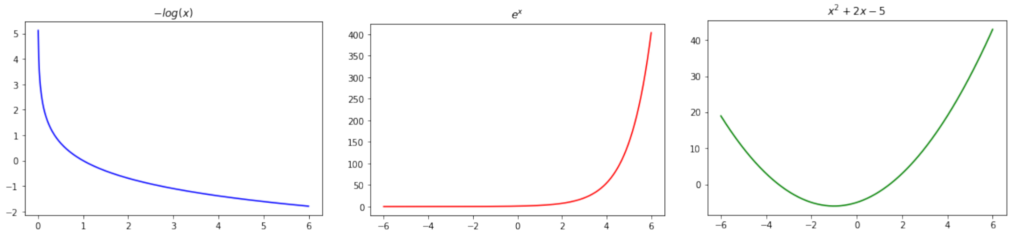


Figure 2.3: Graph of $-log(x)$ (left), e^x (center) and a quadratic function (right)

We now consider the n-dimensional case:

Definition 8. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is self-concordant if*

- *f is convex and three times differentiable*
- *$g(t) = f(x + tv)$ is self-concordant for all $x \in dom(f)$, for all $v \in \mathbb{R}^n$ and $t \in \mathbb{R}$, i.e. its restriction to any arbitrary line is self-concordant.*

The second hypothesis is equivalent to

$$\frac{d}{dt}\nabla^2 f(x+tv)\Big|_{t=0} \preceq 2\|v\|_{\nabla^2 f(x)}\nabla^2 f(x)$$

where $A \preceq B$ means $B - A$ is positive semidefinite.
We list the properties of self-concordant functions:

- stability with respect to affine substitutions: if $f(y)$ is self-concordant then $f(Ax + b)$ is self-concordant;
- stability under summation: if f_1 and f_2 are self-concordant, then $f = f_1 + f_2$ is self-concordant;
- stability under scaling with a positive factor of at least 1: if f is self-concordant and $a > 1$, then af is also self-concordant;
- bounds on hessian: if $x, y \in dom(f)$ and $\|y - x\|_{\nabla^2 f(x)} \leq 1$,

$$(1 - \|y - x\|_{\nabla^2 f(x)})^2 \nabla^2 f(x) \preceq \mathbf{\nabla^2 f(y)} \preceq \frac{1}{(1 - \|y - x\|_{\nabla^2 f(x)})^2}\nabla^2 f(x)$$

- bounds on gradient: if $x, y \in dom(f)$ and $\|x-y\|_{\nabla^2 f(x)} \leq 1$,
$$\|\nabla \mathbf{f}(\mathbf{y}) - \nabla \mathbf{f}(\mathbf{x}) - \nabla^2 \mathbf{f}(\mathbf{x})(\mathbf{y}-\mathbf{x})\|_{\nabla^2 \mathbf{f}(\mathbf{x})^{-1}} \leq \frac{\|y-x\|^2_{\nabla^2 f(x)}}{1-\|y-x\|_{\nabla^2 f(x)}}$$
- bounds on function value: if $x, y \in dom(f)$ and $\|x-y\|_{\nabla^2 f(x)} \leq 1$,
$$w(\|y-x\|_{\nabla^2 f(x)}) \leq \mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x}) - \nabla \mathbf{f}(\mathbf{x})^{\mathbf{T}}(\mathbf{y}-\mathbf{x}) \leq w^*(\|y-x\|_{\nabla^2 f(x)}) \tag{2.18}$$
where w and w^* denote the functions
$$w(u) = u - log(1+u); \quad w^*(u) = -u - log(1-u).$$
Figure 2.4 illustrates the inequality $\frac{u^2}{2(1+u)} \leq w(u) \leq \frac{u^2}{2} \leq w^*(u)$

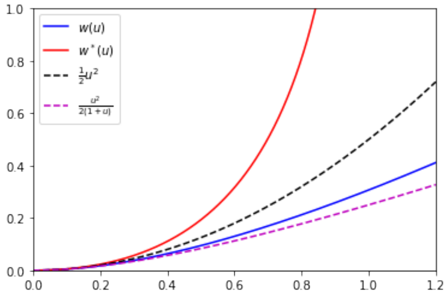


Figure 2.4: The function $w(u) = u - log(1+u)$ and $w^*(u) = -u - log(1-u)$.

- Dikin ellipsoid theorem: the Dikin's ellipsoid of f centered at x of the radius $r < 1$ is
$$W_r(x) = \{y : \|x-y\|_{\nabla^2 f} \leq r\}.$$
If f is self-concordant the upper bound in 2.18 implies that $W_r(x) \in dom(f)$.

Self-concordant functions: proximal Newton method analysis

Now we want to analyze the problem

$$min\{f(x) + g(x)\} \tag{2.19}$$

where f is a self-concordant function and g is a convex funtion with an inexpensive proximal operator. We use the proximal inexact Newton method, the analysis is similar to the previous one, with the main exceptions:

- self-concordance replaces convexity and Lipschitz Hessian assumptions;
- the condition 2.17 is replaced by the following criterion: a step $\tilde{d}$ is accepted as an approximation of d if there exists a residual r such that
$$r \in \nabla f(x) + \nabla^2 f(x)\tilde{d} + \partial g(x+\tilde{d}), \quad \|r\|_{(\nabla^2 f(x))^{-1}} \leq (1-\theta)\|\tilde{d}\|_{\nabla^2 f(x)} \tag{2.20}$$
where $\theta \in (0,1]$ is an algorithm parameter. With $\theta = 1$ the condition requires $r = 0$, therefore we use the exact proximal Newton step.

The next theorem shows the global convergence: if $\tilde{d}$ satisfies 2.20 for some r and $\|\tilde{d}\|_{\nabla^2 f(x)}$ is sufficiently small, then x is close to the optimal solution x^* of 2.19.

Theorem 9. *Suppose $x \in dom(f)$, $x+\tilde{d} \in dom(g)$, $\tilde{d}$ and r satisfy 2.20 with $\theta \in (0,1]$. If*

$$\|\tilde{d}\|_{\nabla^2 f(x)} \leq \frac{1}{2-\theta}$$

then the following properties hold:

1. *f is bounded below and*
$$\inf_y f(y) \geq f(x+\tilde{d}) + \theta\|\tilde{d}\|^2_{\nabla^2 f(x)} - w^*(\|\tilde{d}\|_{\nabla^2 f(x)}) - w^*((2-\theta)\|\tilde{d}\|_{\nabla^2 f(x)})$$
2. *the sublevel set $S_x = \{y : f(y) \leq f(x+\tilde{d})\}$ is bounded: $S_x \subseteq \{y : \|y-x\|_{\nabla^2 f(x)} \leq \rho\}$ where ρ is the positive root of the nonlinear equation*
$$w(\rho) - \rho(2-\theta)\|\tilde{d}\|_{\nabla^2 f(x)} = max\{0, w^*(\|\tilde{d}\|_{\nabla^2 f(x)}) - \theta\|\tilde{d}\|^2_{\nabla^2 f(x)}\}$$
if $\|\tilde{d}\|_{\nabla^2 f(x)} > 0$, and $\rho = 0$ if $\|\tilde{d}\|_{\nabla^2 f(x)} = 0$.
3. *f has a unique minimizer x^* and $\|x - x^*\|_{\nabla^2 f(x)} \leq \rho$.*

In the case of the local convergence the pararmeter $1-\theta$ plays a role similar to the parameter η_k in the previous analysis: we suppose that the hypothesis 2.20 is satisfied then

- the convergence is quadratic if $\theta = 1$;
- the convergence is linear if θ constant and less than one;
- the convergence is superlinear if θ approaches one as the algorithm converges.

2.5 Duality

In some applications it is advantageous to apply a proximal algorithm to the dual problem; in fact if we consider the optimization problem

$$\min_x \{f(x) + g(Ax)\} \tag{2.21}$$

where f is a strongly convex function, g is a convex function and A is a general linear mapping; there is not the hypothesis that the gradient of f is Lipschitz, therefore we can not apply, for example, the proximal gradient method. However if we consider the dual problem we have the desiderable hypothesis. Firstly we give some definitions:

Definition 10. *Let X a real topological vector space, X^* its dual space and let $f : X \to \mathbb{R} \cup \{+\infty\}$ a function. The conjugate convex of f is*

$$f^*(s) = \sup_{x \in X} \{\langle x, s \rangle - f(x)\}$$

Lemma 11. *We discuss some properties of the conjugate f^*:*

1. *f^* is convex, even f is not*

2. *if f is closed and convex then $f^{**} = f$:*

3. *$f^*(s) = s^t x - f(x) \iff s \in \partial f(x) \iff x \in \partial f^*(s)$*

4. *if f is closed and strongly convex with parameter μ, then f^* has a Lipschitz continuos gradient with parameter $1/\mu$*

   Definition 11. *f is strongly convex with parameter $\mu \iff (s_y - s_x)^t (y - x) \geq \mu \|y - x\|^2$, $\forall\ x, y,\ s_x \in \partial f(x),\ s_y \in \partial f(y)$.*

5. *$\nabla f^*(y) = \operatorname{argmax}_x (y^t x - f(x))$*

6. *if f is a positively homogeneous function from $X \to \mathbb{R} \cup \{+\infty\}$, i.e. $f(\lambda x) = |\lambda| f(x)$, then f^* is the indicator function of a closed convex subset K of X^*.*

Proof. 1. The function $h(s) := \langle x, s \rangle - f(x)$ is an affine map, so its epigraph is convex. Now f^* is the pointwise supremum of h and its epigraph is the intersection of the above affine epigraphs. Each epigraph is convex, then the epigraph of f^* is convex $\implies$ f^* is convex.

2. Suppose f is differentiable, then the conjugate function can be obtain as:
$$\nabla_x(s^Tx - f(x)) = 0 \iff s = \nabla_x f(x^*) \quad \exists x^*$$
f is convex, so
$$f^*(s) = \nabla_x f(x^*)^T x^* - f(x^*)$$
$$\begin{aligned} f^{**}(x) = \sup_s(x^Ts - f^*(s)) &= \sup_{x_0}(x^T\nabla f(x_0) - \nabla f(x_0)^t x_0 + f(x_0)) = \\ &= \sup_{x_0}(f(x_0) + \nabla f(x_0)^T(x - x_0)) = f(x) \end{aligned}$$
We used first order condition for a convex function : $f(y) \geq f(x) + \nabla f(x)^T(y - x), \forall x, y$.

3. Firstly we prove the first if and only if:
$$s \in \partial f(x) \iff s^Tx - f(x) \geq s^Ty - f(y) \iff s^Tx - f(x) \geq f^*(s)$$
Since $f^*(s) \geq s^Tx - f(x) \implies f^*(s) = s^Tx - f(x)$.
We prove, now, the second if and only if:
$$\begin{aligned} f^*(z) = \sup_x(z^Tx - f(x)) &\geq z^Tx - f(x) = s^Tx - f(x) + x^T(z - s) \\ &= f^*(s) + x^T(z - s) \end{aligned}$$
Then
$$f^*(z) - x^Tz \geq f^*(s) - x^Ts \implies x \in \partial f^*(s)$$
The other implication follows from $f^{**} = f$.

4. For the definition of strongly convex
$$(s_y - s_x)^t(y - x) \geq \mu\|y - x\|^2 \implies \|s_y - s_x\| \geq \mu\|y - x\|$$
Using the property 3 we have
$$\|s_y - s_x\| \geq \mu\|\nabla f^*(s_y) - \nabla f^*(s_x)\| \implies \|\nabla f^*(s_y) - \nabla f^*(s_x)\| \leq \frac{1}{\mu}\|s_y - s_x\|$$

5. If x is $\max(y^Tx - f(x)) \implies y \in \partial f(x) \iff x \in \partial f^*(y)$. In this way $\operatorname{argmax}(y^Tx - f(x)) = \nabla f^*(y)$.

6. Given f, $f^*(s) = \sup(\langle x, s\rangle - f(x))$ and let $u^* \in X^*$. Two cases occur:

- $\exists x_0 \in X$ such that $\langle x_0, u^*\rangle - f(x_0) > 0 \implies \langle \lambda x_0, u^*\rangle - f(\lambda x_0) = \lambda(\langle x_0, u^*\rangle - f(x_0)) \leq f^*(u^*)$
  Passing to the limit $\lambda \to \infty$ $f^*(u^*)) = +\infty$.
- $\langle x, u^*\rangle - f(x) \leq 0 \; \forall x \in X \implies f^*(u^*) \leq 0$.
  $\langle 0, u^*\rangle - f(0) \leq f^*(u^*)$
  $f(0) = f(n \cdot 0) = nf(0) \; \forall n \in \mathbb{N}$ and f positevely . Now $f(0) = 0 \implies f^*(u^*) = 0$.

Then we define $K = \{u^* \in X^* : f^*(u^*) = 0\}$ and $f^* = \mathbb{I}_K$.

□

We consider now the problem 2.21, we can express it in terms of convex conjugate functions, associating a Langrange variable vector u to the set of equality constraints in

$$\min_{x,z}\{f(x) + g(z) : Ax = z\}$$

So we can construct the Lagrangian of the problem:

$$\begin{aligned}
&\max_u\{\min_{x,z}\{f(x) + g(z) + \langle u, Ax - z\rangle\}\} \\
&\max_u\{\min_{x,z}\{f(x) + \langle A^*u, x\rangle\} + \min_z\{g(z) - \langle u, z\rangle\}\} \\
&\max_u\{-f^*(-A^*u) - g^*(u)\} \implies \min_u\{f^*(-A^*u) + g^*(u)\}
\end{aligned}$$

Observation 1. *the dual problem is useful because by Moreau decomposition whenever the proximal mapping of g is efficiently computable so is that of g^*;*

Observation 2. *for hypothesis f is strongly convex with parameter μ, so f^* has a Lipschitz gradient $\frac{\|A\|^2}{\mu}$: let $F(u) := f^*(-A^*u)$ then (to see Appendix A for computations)*

$$\begin{aligned}
\|\nabla F(x) - \nabla F(y)\| &= \|-A\nabla f^*(-A^*x) + A\nabla f^*(-A^*y)\| \\
&\leq \frac{1}{\mu}\|A\|\|A^*x - A^*y\| \leq \frac{1}{\mu}\|A\|\|A^*\|\|x - y\| \\
&= \frac{1}{\mu}\|A\|^2\|x - y\|
\end{aligned}$$

So we can apply, for example, the proximal gradient method on the dual problem. Let $\tilde{u}$ the optimal solution. It can be converted back to the one of the original problem through $\tilde{x} = \nabla f^*(-A^*\tilde{u})$. In fact applying the PGM:

$$u^{k+1} = prox_{\lambda g^*}(u^k + \lambda A \nabla f^*(-A^* u^k))$$

It admits a more esplicit primal representation:

$$\begin{aligned} x^k &= \nabla f^*(-A^* u^k) \\ u^{k+1} &= prox_{\lambda g^*}(u^k + \lambda A x^k) \end{aligned}$$

If $\tilde{u}$ is optimal solution of the dual and remembered the property 5, we have that $x = \nabla f^*(-A^*\tilde{u}) = \operatorname{argmin}\{f(y) + \langle A^*\tilde{u}, y\rangle\}$ is optimal solution of problem 2.21.

We consider now an example of this approch:

2.5.1 Denoising problem

Many image processing problem can be formulated as estimating the original image $x \in \mathbb{R}^{m\times n}$ from a corrupted observation b, where $b = A(x) + w$ with A a linear operator and w a noise vector. Usually A is ill-conditioned or even singular. Thus image restoration is a classical inverse problem.

To solve the problem we need a regularization term in the objective function

$$\min_{x\in C}\{\varphi(x) = \frac{1}{2}\|A(x) - b\|_F^2 + \lambda\phi(x)\},$$

where $\phi(x)$ is regularizer, λ is a parameter, C is the constraint on the restored image x and $\|\cdot\|_F$ indicates the Frobenius norm.

In this example we consider total variation regularizer and the denoising problem, so A is the operator identity. Our problem becomes:

$$\min_{x\in C}\{\|x - b\|_F^2 + 2\lambda TV(x)\} \tag{2.22}$$

The total variation penalizes large changes in neighboring pixel intensities, making it useful for removing noise. It is defined by

$$TV(x) = \sum_{i=1}^{m}\sum_{j=1}^{n}(|x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|)$$

with the conditions

$$x_{m+1,j} = x_{m,j} \quad \text{for} \quad j = 1, ..., n$$

$$x_{i,n+1} = x_{i,n} \quad \text{for} \quad i = 1, ..., m$$

Now we want to write the dual problem of 2.22. We introduce the following notation:

- the linear operator $\mathcal{L} : \mathbb{R}^{(m-1)\times n} \times \mathbb{R}^{m\times(n-1)} \to \mathbb{R}^{m\times n}$ is defined by

  $$\mathcal{L}(p, q)_{i,j} = p_{i,j} + q_{i,j} - p_{i-1,j} - q_{i,j-1}, \quad i = 1, ..., m, j = 1, ..., n$$

  with the condition $p_{0,j} = p_{m,j} =$ for $j = 1, ..., n$, $q_{i,0} = q_{i,n} = 0$ for $i = 1, ..., m$;

- the operator $\mathcal{L}^* : \mathbb{R}^{m\times n} \to \mathbb{R}^{(m-1)\times n} \times \mathbb{R}^{m\times(n-1)}$, which is the adjoint of $\mathcal{L}$, is defined by

  $$\mathcal{L}^*(x) = (p, q) \quad \text{with} \quad p_{i,j} = x_{i,j} - x_{i+1,j}, \quad q_{i,j} = x_{i,j} - x_{i,j+1}$$

Now the total variation is a positively homogeneous function, then for the property 6 of the conjugate function , its coniugate function is the indicator function of a closed convex set K. Moreover $TV^{**} = TV$, so

$$TV(x) = \sup_{v\in K}\langle x, v\rangle$$

We remember the definition of total variation for a function f in $\Omega \subset \mathbb{R}^n$, belonging to L^1:

$$TV(f) = \sup\left\{\int_\Omega f(x) div\phi(x)\, dx : \phi \in C^1_C(\Omega), \|\phi\|_{L^\infty(\Omega)} \leq 1\right\}$$

where $C^1_C(\Omega)$ is the set of continuously differentiable functions of compact support contained in Ω. Then we note the relation:

$$|x| + |y| = \max_{p,q}\{p|x| + q|y| : |p| \leq 1, |q| \leq 1\}$$

Hence we can write

$$\begin{aligned} TV(x) &= \sup_{p,q}\langle x,(p,q)\rangle = \max_{p,q}\sum_{i=1}^{m}\sum_{j=1}^{n}(p_{i,j}|x_{i+1,j}-x_{i,j}|+q_{i,j}|x_{i,j+1}-x_{i,j}|) \\ &= \max_{p,q} Tr(\mathcal{L}(p,q)^t x)) \end{aligned} \tag{2.23}$$

where $|p_{i,j}| \leq 1$ for $i = 1, ..., m-1$, $j = 1, ..., n$; $|q_{i,j}| \leq 1$ for $i = 1, ..., m$, $j = 1, ..., n-1$.

Theorem 12. *Let $(\tilde{p},\tilde{q})$ the optimal solution of the problem*

$$\min_{p,q}\{h(p,q) = -\|H_C(b-\lambda\mathcal{L}(p,q))\|_F^2 + \|b-\lambda\mathcal{L}(p,q)\|_F^2\}$$

where $H_C(x) = x - P_C(x)$ and $P_C(x)$ is the projection on the set C. Then the optimal solution of 2.22 is given by

$$\tilde{x} = P_C(b-\lambda\mathcal{L}(\tilde{p},\tilde{q})).$$

Proof. The problem 2.22, using 2.23, becomes

$$\min_{x\in C}\max_{p,q}\{\|x-b\|_F^2 + 2\lambda Tr(\mathcal{L}(p,q)^t x))\}$$

We exchange the order of the minimum and maximum and we get:

$$\max_{p,q}\min_{x\in C}\{\|x-b\|_F^2 + 2\lambda Tr(\mathcal{L}(p,q)^t x))\}$$

which can be rewritten as

$$\max_{p,q}\min_{x\in C}\{\|x-(b-\lambda\mathcal{L}(p,q))\|_F^2 - \|b-\lambda\mathcal{L}(p,q)\|_F^2 + \|b\|_F^2\} \tag{2.24}$$

So the optimal solution of the problem is

$$\tilde{x} = P_C(b-\lambda\mathcal{L}(p,q)).$$

Now we plug the expression for x in 2.24 and we obtain

$$\max_{p,q}\{\|P_C(b-\lambda\mathcal{L}(p,q)) - (b-\lambda\mathcal{L}(p,q))\|_F^2 + \|b-\lambda\mathcal{L}(p,q)\|_F^2\}$$

□

To find $(\tilde{p}, \tilde{q})$, the optimal solution of the dual problem, we can use PGM, infact $h(p, q)$ is continuosly differentiable and has Lipschitz gradient. Firstly we prove that

$$\nabla h(p, q) = -2\lambda \mathcal{L}^* P_C(b - \lambda \mathcal{L}(p, q)).$$

To ease the notation we set $s : \mathbb{R}^{m \times n} \to \mathbb{R}$ defined by $s(x) = \|H_C(x)\|_F^2$. So $\nabla s(x) = 2(x - P_C(x))$.

$$\begin{aligned}
\nabla h(p, q) &= \lambda \mathcal{L}^* \nabla s(b - \lambda \mathcal{L}(p, q)) - 2\lambda \mathcal{L}^*(b - \lambda \mathcal{L}(p, q)) \\
&= 2\lambda \mathcal{L}^*(b - \lambda \mathcal{L}(p, q)) - 2\lambda \mathcal{L}^* P_C(b - \lambda \mathcal{L}(p, q)) - 2\lambda \mathcal{L}^*(b - \lambda \mathcal{L}(p, q)) \\
&= -2\lambda \mathcal{L}^* P_C(b - \lambda \mathcal{L}(p, q)).
\end{aligned}$$

Therefore we can prove thata ∇h is Lipschitz:

$$\begin{aligned}
\|\nabla h(p_1, q_1) - \nabla h(p_2, q_2)\| &= 2\lambda \|\mathcal{L}^* P_C(b - \lambda \mathcal{L}(p_1, q_1)) - \mathcal{L}^* P_C(b - \lambda \mathcal{L}(p_2, q_2))\| \\
&\leq 2\lambda \|\mathcal{L}^*\| \|P_C(b - \lambda \mathcal{L}(p_1, q_1)) - P_C(b - \lambda \mathcal{L}(p_2, q_2))\| \\
&\leq 2\lambda^2 \|\mathcal{L}^*\| \|\mathcal{L}(p_1, q_1) - \mathcal{L}(p_2, q_2)\| \\
&\leq 2\lambda^2 \|\mathcal{L}^*\| \|\mathcal{L}\| \|(p_1, q_1) - (p_2, q_2)\| \\
&= 2\lambda^2 \|\mathcal{L}^*\|^2 \|(p_1, q_1) - (p_2, q_2)\|
\end{aligned}$$

So using the notation of the previous section we have

- $f(x) = \|x - b\|_F^2 + \mathbb{I}_C$;
- $g(x) = 2\lambda TV(x)$;
- $f^*(-A^*(p, q)) = h(p, q)$:
- $g^*(p, q) =$ indicator function of the set of (p, q).

Chapter 3

System identification

In this chapter we talk about a subspace method in system identification based on nuclear norm approximation. The classical subspace algorithms rely on singular value decompositions (SVD) and they are efficient methods for making low-rank approximations of matrices constructed from the observed inputs and outputs. However, in this way, the structure of the matrices is lost and the estimate of the range of the extended observability matrix is not always optimal. Moreover the presence of SVD makes it difficult to extend the subspace methods to problems with missing input or output measurement data, to incorporate prior knowledge (for example bounds on the outputs) or to add regularitation terms on the model.

For these reasons minimizing the nuclear norm provides an alternative. In this new method the nuclear norm approximation is used as a pre-processing step, that computes a modified output sequence which is passed to the standard subspace system identification algorithms. We use the proximal algorithm to solve nuclear norm optimization problems.

In the first part of this chapter we review the most common subspace identification algorithm and, then, we formulate nuclear norm variants of these methods.

3.1 Subspace system identification

A linear time-invariant system with discrete time (DLTI) is a operator T which transforms a sequence with discrete time $u(:)$, inputs, into a sequence with discrete time $y(:)$, called outputs: (the symbol " : " indicates that we consider all the indices)

$$y(:) = T[u(:)]$$

It has the following properties:

- linearity: if $y_1(:)$ and $y_2(:)$ are the outputs of the inputs $u_1(:)$ and $u_2(:)$, then

$$T[\alpha_1 u_1(:) + \alpha_2 u_2(:)] = \alpha_1 T[u_1(:)] + \alpha_2 T[u_2(:)] = \alpha_1 y_1(:) + \alpha_2 y_2(:)$$

- time-invariance, i.e. the operator is invariant to shifts along the time axis: shifting the input sequence $u(: -j)$ causes a corresponding shift in the output sequence $y(: -j)$.

To describe the internal dynamics of the system, we introduce a set of internal variables, called state vector. In this way we have a state-space representation.

Let $x(t) \in \mathbb{R}^n$ a vector as function of the time, $u(t) \in \mathbb{R}^{n_u}$ the vector of inputs and $y(t) \in \mathbb{R}^{n_y}$ the vector of outputs. Then, the state-space representation of a continuous DLTI system is the following:

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t) + D_c u(t) \end{aligned}$$

where A_c, B_c, C_c and D_c are matrices. The number of components of the state vector is called order of the model.

To work with the calculator, we write an equivalent discrete time system through discretization:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^{n_u}$ and $y(k) \in \mathbb{R}^{n_y}$ are, respectively, the state vector, the vector of inputs and of outputs at the instant k.

The Markov coefficients G_k are defined by:

$$G_0 = D, \quad G_k = CA^{k-1}B \quad k = 1, 2, ...$$

and they are the discrete time impulse response, i.e. they are the outputs of the model if the i-th inputs are the discrete impulse

$$\delta(k) = \begin{cases} 1 & \text{if} \quad k = 0 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$h^{(i)}(0) = D, \quad h^{(i)}(k) = CA^{k-1}B \quad k = 1, 2, ... \tag{3.1}$$

Given a sequence of observed inputs $u(k)$ and outputs $y(k)$, our objective is:

- to find the minimum model order;
- to estimate the system matrices (A, B, C, D).

Namely, we want to find the minimum realization.

Definition 13. *(A, B, C, D) is a realization of the sequence $\{G_k\}_k$ if it is worth the 3.1.*

Definition 14. *The realization (A, B, C, D) is minimum if model order is minimum.*

Theorem 15. *A realization is minimum if and only if it is reachable and observable.*

Definition 16. *Given a state-space system $\Sigma = (A, B, C, D)$, a state $\bar{x}$ is reachable from the state 0 if there exists an input sequence $\bar{u}(k)$, with finite power, and an instant $\bar{k} < \infty$ such that*

$$\bar{x} = \mathcal{T}(\bar{u}(k), 0, \bar{k})$$

where $\mathcal{T}(u, x_0, k)$ is the trajectory defined by

$$\mathcal{T}(u, x_0, k) = A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} B u(j) \quad k \geq 0.$$

Definition 17. *Given a state-space system $\Sigma = (A, B, C, D)$, two states x^I and x^{II} are indistinguishable in the future in k steps if for every input sequence $u(i)$, $i = 0, .., k-1$ the respective output sequence $y^I(k)$ and $y^{II}(k)$, obtained by the initial states x^I and x^{II}, coincide in the first k steps.*

A state $\bar{x}$ is not observable if it is indistinguishable in the future from the state zero.

So, to find the minimum realization from a sequence of observed inputs $u(k)$ and outputs $y(k)$, we use a subspace method. Briefly we review the basic ideas of subspace identification method. We notice that

$$\begin{aligned} y(k+1) &= Cx(k+1) + Du(k+1) \\ &= CAx(k) + CBu(k) + Du(k+1) \end{aligned}$$

With a recursive replacement we obtain

$$\begin{bmatrix} y(k) \\ \vdots \\ y(k+r) \end{bmatrix} = \begin{bmatrix} C \\ \vdots \\ CA^r \end{bmatrix} x(k) + \begin{bmatrix} D & 0 & \dots & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ CA^{r-1}B & \dots & \dots & CB & D \end{bmatrix} \begin{bmatrix} u(k) \\ \vdots \\ u(k+r) \end{bmatrix}$$

If we indicate with $H_{i,j,k}$ the $j \times k$ block Hankel matrix,

$$H_{i,j,k} = \begin{bmatrix} h(i) & h(i+1) & h(i+2) & \dots & h(i+k-1) \\ h(i+1) & h(i+2) & h(i+3) & \dots & h(i+k) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(i+j-1) & h(i+j) & h(i+j+1) & \dots & h(i+j+k-2) \end{bmatrix}$$

the starting point of subspace method is the matrix equation

$$Y_{0,r,N} = O_r X_{0,1,N} + S_r U_{0,r,N}$$

The matrices $Y_{0,r,N}$ and $U_{0,r,N}$ are block Hankel matrices constructed from the sequances $y(k)$, $u(k)$, for $k = 0, ..., r+N-2$, the matrix $X_{0,1,N}$ has as its columns the states $x(k)$, $k = 0, .., N-1$ and the matrices O_r and S_r contain all model parametres (A, B, C, D). The matrix O_r is called extended observability matrix and the matrix S_r contains the Markov coefficents:

$$Y_{0,r,N} = \begin{bmatrix} y(0) & y(1) & y(2) & \dots & y(N-1) \\ y(1) & y(2) & y(3) & \dots & y(N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(r-1) & y(r) & y(r+1) & \dots & y(N+r-2) \end{bmatrix}$$

$$U_{0,r,N} = \begin{bmatrix} u(0) & u(1) & u(2) & \dots & u(N-1) \\ u(1) & u(2) & u(3) & \dots & u(N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u(r-1) & u(r) & u(r+1) & \dots & u(N+r-2) \end{bmatrix}$$

$$X_{0,1,N} = \begin{bmatrix} x(0) & x(1) & x(2) & \dots & x(N-1) \end{bmatrix}$$

$$O_r = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{r-1} \end{bmatrix} \qquad S_r = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & & & \ddots & \\ CA^{r-2}B & CA^{r-3}B & \dots & CB & D \end{bmatrix}$$

The susbspace method first estimates the range space of the extended observability matrix O_r and, then, determines a system realization from the estimate of $range(O_r)$. Therefore, the method requires that r is taken greater than n.

3.1.1 Extended observability matrix

In the simplest variant of the subspace methods, the matrix $Y_{0,r,N}$ is multiplied on the right with a projection matrix that projects on the nullspace of $U_{0,r,N}$. In this way the term $S_rU_{0,r,N}$ disappears. This gives the equation

$$\begin{aligned} Y_{0,r,N}\Pi_{0,r,N} &= O_rX_{0,1,N}\Pi_{0,r,N} + S_rU_{0,r,N}\Pi_{0,r,N} \\ &= O_rX_{0,1,N}\Pi_{0,r,N} \end{aligned}$$

where $\Pi_{0,r,N}$ is the orthogonal projection matrix on the nullspace of $U_{0,r,N}$. Let's see how it's done $\Pi_{0,r,N}$. Let A and B two matrices and we want to project A orthogonally to the row space of B, indicated with A/B. So

$$A/B = CB \quad \text{with} \quad C = \underset{M}{\text{argmin}}\|A - MB\|.$$

Now the projection is orthogonal, then $A - A/B \perp$ row space of B:

$$(A - CB)B^T = 0 \quad \implies \quad CBB^T = AB^T \quad \implies \quad C = AB^T(BB^T)^{-1}$$

Then:

$$A/B = A\Pi_B, \quad \text{with} \quad \Pi_B = B^T(BB^T)^{-1}B$$

Now $A - A/B = A - A\Pi_B = A(I - \Pi_B) = A\Pi_B^{\perp}$, so $\Pi_B^{\perp}$ is the projection to the orthogonal complement of B.

Hence, in our case, the orthogonal projection matrix on the nullspace of $U := U_{0,r,N}$ is

$$\Pi_{0,r,N} = \Pi_U^{\perp} = I - U^T(UU^T)^{-1}U.$$

In fact $U\Pi_U^{\perp} = U(I - U^T(UU^T)^{-1}U) = 0$. We notice that the matrix $\Pi_{0,r,N}$ depends only on the inputs; in this way the left side of the equation depends on both the ouputs and the inputs. So, to estimate the rank and the range of the extended observability matrix we can use the left side, in fact from the observations we have both inputs and outputs.

$$Y_{0,r,N}\Pi_{0,r,N} = O_r X_{0,1,N}\Pi_{0,r,N} \tag{3.2}$$

However, the range of $Y_{0,r,N}\Pi_{0,r,N}$ does not necessarily converge to the range of O_r as N, the number of data, goes to infinity. This deficiency can be resolved by the use of instrumental variables. We define an instrumental variable matrix

$$\Phi = \begin{bmatrix} U_{-s,s,N} \\ Y_{-s,s,N} \end{bmatrix}$$

by combining Hankel matrices of past inputs and outputs. Multiplying 3.2 on the right with Φ^T gives

$$Y_{0,r,N}\Pi_{0,r,N}\Phi^T = O_r X_{0,1,N}\Pi_{0,r,N}\Phi^T$$

It can be shown that [1].

- $\lim_{N\to\infty} \frac{1}{N} X_{0,1,N}\Pi_{0,r,N}\Phi^T$ has full rank n;
- the range of $Y_{0,r,N}\Pi_{0,r,N}\Phi^T$ gives a consistent estimate of range of O_r.

From a practical point of view, for finite N, a truncated SVD of $Y_{0,r,N}\Pi_{0,r,N}\Phi^T$ is used to estimate $range(O_r)$. In fact, firstly, we compute an LQ factorization of the matrix of the stacked input abd output Hankel matrices:

[1] Further details: M.Varhaegen,V. Verdult, "Filtering and System Identification", Cambridge University Press, New York, 2007.

$$\begin{bmatrix} U_{0,r,N} \\ \Phi \\ Y_{0,r,N} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

The matrices L_{11}, L_{22} and L_{33} are triangular matrices of order rn_u, $s(n_u + n_y)$ and rn_y, while the matrices Q_i have column dimension N and satisfy the orthogonal properties: $Q_i Q_i^t = I$ and $Q_i Q_j^T = 0$ for $i \neq j$. Now

$$\Pi_{0,r,N} = I - (L_{11}Q_1)^T [L_{11}Q_1(L_{11}Q_1)^T]^{-1}(L_{11}Q_1) = I - Q_1^T Q_1$$

Then:

$$\begin{aligned} Y_{0,r,N}\Pi_{0,r,N}\Phi^T &= (L_{31}Q_1 + L_{32}Q_2 + L_{33}Q_3)(I - Q_1^T Q_1)(L_{21}Q_1 + L_{22}Q_2)^T \\ &= (L_{31}Q_1 + L_{32}Q_2 + L_{33}Q_3)Q_2^T L_{22}^T \\ &= L_{32}L_{22}^T \end{aligned}$$

The accurancy of subspace method can be improved by multiplying the matrix $L_{32}L_{22}^T$ on both sides with nonsingular matrices before computing SVD. So if

$$G = W_1 L_{32} L_{22}^T W_2 \tag{3.3}$$

after truncating SVD

$$G = \begin{bmatrix} P & P_e \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma_e \end{bmatrix} \begin{bmatrix} Q & Q_e \end{bmatrix},$$

by discarding the smallest singular value Σ_e, we obtain:

- n is the number of non-null singular values of Σ;
- $range(O_r) \approx range(W_1^{-1}P)$.

There are different choices of the weight matrices W_1 and W_2, for example:

- $W_1 = I$, $W_2 = (\Phi\Pi_{0,r,N}\Phi^T)^{-1/2}$;
- $W_1 = (Y_{0,r,N}\Pi_{0,r,N}Y_{0,r,N}^T)^{-1/2}$, $W_2 = (\Phi\Phi^T)^{-1/2}$.

3.1.2 System realization

Once an estimate of $range(O_r)$ has been determined as described in the previous section, we can calculate a system realization and an estimate of the initial state.

We remember that the extended observability matrix is defined by

$$O_r = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{r-1} \end{bmatrix}$$

So, let $V \in \mathbb{R}^{rn_y \times n}$ be a matrix whose columns form a basis of our estimate of $range(O_r)$. Partition V in r block rows $V_0, ..., V_{r-1}$ of size $n_y \times n$:

$$V = \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_{r-1} \end{bmatrix}$$

Then one can take as estimates of C and A the matrices:

$$\hat{C} = V_0, \qquad \hat{A} = \underset{\hat{A}}{\operatorname{argmin}} \sum_{i=1}^{r-1} \|V_i - V_{i-1}\hat{A}\|_F^2$$

where $\|\cdot\|_F$ denotes the Frobenius norm. From $\hat{C}$ and $\hat{A}$, we estimate B, D and $x(0)$ solving a least-squares problem; remembering the equation

$$y(r) = CA^r x(0) + \sum_{i=0}^{r-1} CA^{r-1-i}Bu(i) + Du(r)$$

we have

$$(\hat{B}, \hat{D}, \hat{x}_0) = \underset{\hat{B},\hat{D},\hat{x}_0}{\operatorname{argmin}} \sum_{k=0}^{N+r-2} \|\hat{C}\hat{A}^k\hat{x}_0 + \sum_{i=0}^{k-1} \hat{C}\hat{A}^{k-1-i}\hat{B}u(i) + \hat{D}u(k) - y(k)\|_2^2$$

3.2 Identification by nuclear norm optimization

In this section, we discuss several variations of the subspace methods of previous section based on the minimization of the nuclear norm. We focus on applications to identification with missing data.

We look for modified outputs which are close to the measurement values and such that at the optimum the matrix $G(\boldsymbol{y})$ is low rank. The desired result is that the matrix $G(\boldsymbol{y})$ has a minimum rank because its rank is equal to the order of the model and our objective is to find the minimum model order.

The idea of replacing the rank of a matrix by its nuclear norm can be justified as a convex relaxation (the nuclear norm is the largest convex lower bound of $rank(A)$ on the ball $\{A : \|A\|_2 = \sigma_1(A) \leq 1\}$. It is further motivated by the empirical observation that the minimum nuclear norm solutions often have low rank. [2] [3].

Definition 18. *Let A a matrix in $\mathbb{R}^{m\times n}$, the nuclear norm (or trace norm) of A is defined as*

$$\|A\|_* = trace(\sqrt{A^T A}) = \sum_{i=1}^{min\{m,n\}} \sigma_i(A)$$

where $\sigma_i(A)$ denotes the singular value of matrix A.

We first consider an identification problem with complete data. Let $y_{meas}(k)$ and $u_{meas}(k)$ the measured data. The model outputs $y(k)$ are computed by solving a regularized nuclear norm problem

$$minimize\|G(\boldsymbol{y})\|_* + \lambda \sum_{k\in T} \|y(k) - y_{meas}(k)\|_2^2 \tag{3.4}$$

The optimization variable is the sequence $\boldsymbol{y} = (y(0), y(1), ..., y(N+r-2))$. We have that

[2] M. Fazel, H. Hindi, S. Boyd, "A rank minimization heuristic with application to minimum order system approximation", Proceedings og the American Control Conference, 2001, pp.4734-4739

[3] M. Fazel,"Matrix rank minimization with applications", Ph.D. thesis, Stanford University (2002)

- $$G(\boldsymbol{y}) = W_1 Y_{0,r,N} \Pi_{0,r,N} \Phi^T W_2$$

  where we use the measured inputs and outputs to construct W_1, W_2, $\Pi_{0,r,N}$ and Φ, and define $Y_{0,r,N}$ as the Hankel matrix constructed from the model outputs $y(k)$;

- λ is a positive weight;

- the second term in the objective function measures the deviation between the computed model outputs and the measurement data;

- the index set $T = \{0, 1, ..., N + r - 2\}$.

So, the problem 3.4 tries to find values of the outputs that are close to the measurement values and make the matrix $G(\boldsymbol{y})$ low-rank. We notice that we do not guarantee that we minimize the rank of $G(\boldsymbol{y})$. After having computed $\boldsymbol{y}$, we use the matrix $G(\boldsymbol{y})$ as G in 3.3 to obtain an estimation of the range of O_r and, then, we proceed with a system realization as described in the previous section.

Now, we can extend the formulation to problems with part of the measured outputs is missing. In this case, the aim of the nuclear norm minimization is to complete the output sequence. Compared to the preceded formulation, we have the following differences:

- T is defined as the set of indices from which $y_{meas}(k)$ is available;

- we exclude the outputs from the instrumental variable and use $\Phi = U_{-s,s,N}$, instaed of $\Phi = \begin{bmatrix} U_{-s,s,N} \\ Y_{-s,s,N} \end{bmatrix}$;

- there are also restrictions in the choices of the weight matrices; for example we can not use the matrix $W_1 = (Y_{0,r,N} \Pi_{0,r,N} Y_{0,r,N}^T)^{-1/2}$ because it requires complete outputs.

So our problem is

$$\begin{array}{ll} minimize & \|G(\boldsymbol{y})\|_* \\ s.t. & y(k) = y_{meas}(k), \quad k \in T \end{array}$$

which is equivalent to the problem

$$minimize\|G(\boldsymbol{y})\|_* + \lambda \sum_{k \in T} \|y(k) - y_{meas}(k)\|_2^2$$

with variables $\boldsymbol{y} = (y(0), ..., y(N+r-2))$ to estimate corrected values of the measured outputs and simultaneously estimate the missing outputs.

3.3 Solution of nuclear norm problem via proximal algorithms

In this section we solve the nuclear norm optimization problem 3.4 using the proximal algorithms of Chapter 2. We can express the problem in the following general form, a generic nuclear norm optimization problem with a quadratic regularization term:

$$minimize \quad \|\mathcal{A}(x)\|_* + \frac{1}{2}(x-a)^T E(x-a) \tag{3.5}$$

The variables is a vector $x \in \mathbb{R}^n$. The matrix $\mathcal{A}(x) \in \mathbb{R}^{p \times q}$ where $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^{p \times q}$ is a linear mapping, $a \in \mathbb{R}^n$ is a vector and E is a positive semidefinite symmetric matrix. So in our case x is formed by stacking columns of y one by one, a is formed by stacking columns of y_{meas}, E is the identity matrix, we substitute 1/2 with λ and $\mathcal{A}(x) = G(\boldsymbol{y})$.

3.3.1 ADMM algorithm

To derive the ADMM iteration, we write 3.5 as

$$\begin{aligned} &minimize \quad \|X\|_* + \frac{1}{2}(x-a)^T E(x-a) \\ &s.t. \qquad \mathcal{A}(x) = X \end{aligned}$$

and we use the interpretation of ADMM like augmented Lagrangian. The augmented Lagrangain for this problem is

$$L_\rho(x, X, Z) = \|X\|_* + \frac{1}{2}(x-a)^T E(x-a) + Tr(Z^T(\mathcal{A}(x) - X)) + \frac{\rho}{2}\|\mathcal{A}(x) - X\|_F^2$$

where ρ is a positive penalty parameter, $x \in \mathbb{R}^n$, $X \in \mathbb{R}^{p\times q}$ and $Z \in \mathbb{R}^{p\times q}$. We saw in Chapter 2 that each iteration consists of a minimization of L_ρ over x, a minimization over X and an update of the dual variable Z.

```
**initialize**  $x, X, Z, \rho$  For example, set  $x = 0, X = 0, Z = 0, \rho = 1$
1.  **update**  $x = \operatorname*{argmin}_{\hat{x}} L_\rho(\hat{x}, X, Z)$;
2.  **update**  $X = \operatorname*{argmin}_{\hat{X}} L_\rho(x, \hat{X}, Z)$;
3.  **update**  $Z = Z + \rho(\mathcal{A}(x) - X)$;
**repeat**  until stopping conditions are satisfied
```

Minimizer over X

The minimizer X in step 2. can be expressed as

$$\begin{aligned}
X &= \operatorname*{argmin}_{\hat{X}} L_\rho(x, \hat{X}, Z) \\
&= \operatorname*{argmin}_{\hat{X}} \left\{ \|\hat{X}\|_* + \frac{1}{2}(x-a)^T E(x-a) + Tr(Z^T(\mathcal{A}(x) - \hat{X})) + \frac{\rho}{2}\|\mathcal{A}(x) - \hat{X}\|_F^2 \right\} \\
&= \operatorname*{argmin}_{\hat{X}} \left\{ \|\hat{X}\|_* - Tr(Z^T\hat{X}) + \frac{\rho}{2}\|\mathcal{A}(x) - \hat{X}\|_F^2 \right\} \\
&= \operatorname*{argmin}_{\hat{X}} \left\{ \|\hat{X}\|_* + \frac{\rho}{2}\|\hat{X} - \mathcal{A}(x) - \frac{1}{\rho}Z\|_F^2 \right\} \\
&= prox^F_{\frac{\|\cdot\|_*}{\rho}}\left(\mathcal{A}(x) + \frac{1}{\rho}Z\right)
\end{aligned}$$

where $prox^F_{\frac{\|\cdot\|_*}{\rho}}(\cdot)$ is similar to the proximal operator of Chapter 1, with the difference that the norm in the definition is not the Euclidean norm but the Frobenius norm because there are matrices. Now, we will see how to compute the proximal operator of a nuclear norm:

- we apply the SVD on $\mathcal{A}(x)+\frac{1}{\rho}Z \rightarrow U\Sigma V^T$, whre Σ is a diagonal matrix with in the diagonal the singular values of $\mathcal{A}(x) + \frac{1}{\rho}Z$;
- we extract the vector of the singular values from $\Sigma = diag(\sigma_i)$;

- we compute the proximal operator of the extracted vector using $\|\cdot\|_1$

$$\begin{aligned}\hat{\sigma_i} &= prox_{\frac{\|\cdot\|_1}{\rho}}(\sigma_i) = \operatorname*{argmin}_x \{\|x\|_1 + \frac{\rho}{2}\|x-\sigma_i\|_2^2\} \\ &= \begin{cases} \sigma_i - \frac{1}{\rho} & \text{if } \sigma_i \geq \frac{1}{\rho} \\ 0 & \text{if } -\frac{1}{\rho} \leq \sigma_i \leq \frac{1}{\rho} \\ \sigma_i + \frac{1}{\rho} & \text{if } \sigma_i \leq -\frac{1}{\rho} \end{cases} \\ &= \max\left\{0, \sigma_i - \frac{1}{\rho}\right\}\end{aligned}$$

  where the last equality is true because the singular values are all non-negative;

- we return to the proximal operator of the matrix norm

$$\begin{aligned}X &= prox^F_{\frac{\|\cdot\|_*}{\rho}}\left(\mathcal{A}(x) + \frac{1}{\rho}Z\right) \\ &= Udiag(\hat{\sigma}_i)V^T \\ &= \sum_i^{\min\{p,q\}} \max\left\{0, \sigma_i - \frac{1}{\rho}\right\} u_i v_i^T\end{aligned}$$

Minimizer over x

While minimizing L_ρ with respect to $\hat{X}$ admits an easy closed form solution, minimizing L_ρ with respect to $\hat{x}$ does not usually have a simple closed form solution due to the quadratic terms

The update in step 1 is:

$$\begin{aligned}x &= \operatorname*{argmin}_{\hat{x}} L_\rho(\hat{x}, X, Z) \\ &= \operatorname*{argmin}_{\hat{x}} \left\{\|X\|_* + \frac{1}{2}(\hat{x}-a)^T E(\hat{x}-a) + Tr(Z^T(\mathcal{A}(\hat{x}) - X)) + \frac{\rho}{2}\|\mathcal{A}(\hat{x}) - X\|_F^2\right\} \\ &= \operatorname*{argmin}_{\hat{x}} \left\{\frac{1}{2}(\hat{x}-a)^T E(\hat{x}-a) + Tr(Z^T\mathcal{A}(\hat{x})) + \frac{\rho}{2}\|\mathcal{A}(\hat{x}) - X\|_F^2\right\} \\ &= \operatorname*{argmin}_{\hat{x}} \left\{\frac{1}{2}(\hat{x}-a)^T E(\hat{x}-a) + \frac{\rho}{2}\|\mathcal{A}(\hat{x}) - X + \frac{1}{\rho}Z\|_F^2\right\}\end{aligned}$$

One way to solve this is to compute the solution of a linear equation, in fact setting the gradient of $L_\rho(\hat{x}, X, Z)$ with respect to $\hat{x}$ equal to zero gives the equation:

$$\begin{aligned}
&E(\hat{x}-a)+\frac{\rho}{2}\left(2\mathcal{A}_{adj}\mathcal{A}(\hat{x})-2\mathcal{A}_{adj}X+\frac{2}{\rho}\mathcal{A}_{adj}Z\right)=0\\
&E(\hat{x}-a)+\rho\mathcal{A}_{adj}\mathcal{A}(\hat{x})-\rho\mathcal{A}_{adj}X+\mathcal{A}_{adj}Z=0\\
&(E+\rho M)\hat{x}=\mathcal{A}_{adj}(\rho X-Z)+Ea
\end{aligned}$$

where $\mathcal{A}_{adj}$is the adjoint of the mapping $\mathcal{A}$ and M is the positive semidefinite matrix defined by the identity

$$Mz=\mathcal{A}_{adj}(\mathcal{A}(z)) \quad \forall z$$

We exploit the Hankel structure in the subspace system identification applications. The mapping $\mathcal{A}$ can be expressed as

$$\mathcal{A}(x)=L\mathcal{H}(x)R$$

where $x=(h_1,...,h_{r+N-1})$ with $h_i\in\mathbb{R}^u$, and $\mathcal{H}(x)$ is a block Hankel matrix.

$$\mathcal{H}(x)=\begin{bmatrix} h_1 & h_2 & \dots & h_N\\ h_2 & h_3 & \dots & h_{N+1}\\ \vdots & \vdots & \ddots & \vdots\\ h_r & h_{r+1} & \dots & h_{r+N-1}\end{bmatrix}$$

For example, the matrix $G(\boldsymbol{y})$ in the nuclear norm identification problem can be written in this form with

- $L=W_1$;
- $\mathcal{H}(x)=Y_{0,r,N}$;
- $R=\Pi_{0,r,N}\Phi^T W_2$;
- $x=(y(0),...,y(r+N-2))$.

Now the adjoint of the mapping $\mathcal{A}$ is $\mathcal{A}_{adj}(Y)=\mathcal{H}_{adj}(L^T Y R^T)$. The adjoint $\mathcal{H}_{adj}$ of the Hankel mapping $\mathcal{H}$ maps an $ru\times N$ matrix to a sequence of $r+N-1$ vectors of size u by summing the block entries in the matrix along the anti-diagonals. So, for example, if X is an $r\times N$ block matrix with block

$x_{ij} \in \mathbb{R}^u$, then $\mathcal{H}_{adj}(X) = (y_1, ..., y_{r+N-1})$ with $y_k = \sum_{i+j=k+1} x_{ij}$. For any further detail, please read Appendix A.

An improvement in the algorithm is to avoid the inverse in $\hat{x} = (E + \rho M)^{-1}(\mathcal{A}_{adj}(\rho X - Z) + Ea)$ by introducing an additional proximal quadratic term to augmented Lagrangian so it "cancels" out the term $E + \rho M$. The idea is similar to the linearized ADMM that we saw in Chapter 2. In this approach, we update (we indicate with x^k the k-th iteration):

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \left\{ L_\rho(x, X^k, Z^k) + \frac{\rho}{2} \|x - x^k\|_Q^2 \right\}$$

with

$$Q = \left(s + \frac{1}{\rho}\right) I - \left(\mathcal{A}_{adj}\mathcal{A} + \frac{1}{\rho}E\right) \succeq 0, \quad s = min\{r, N\}$$

We set $\delta := \left(s + \frac{1}{\rho}\right)$,

$$0 = Ex^{k+1} - Ea + \rho\mathcal{A}_{adj}\mathcal{A}(x^{k+1}) - \rho\mathcal{A}_{adj}X^k + \mathcal{A}_{adj}Z^k + \rho[\delta(x^{k+1} - x^k) - \mathcal{A}_{adj}\mathcal{A}(x^{k+1} - x^k) - \frac{1}{\rho}E(x^{k+1} - x^k)]$$

$$\begin{aligned} 0 &= -Ea - \rho\mathcal{A}_{adj}X^k + \mathcal{A}_{adj}Z^k + \rho\delta x^{k+1} - \rho\delta x^k + \rho\mathcal{A}_{adj}\mathcal{A}(x^k) + \rho x^k \\ \rho\delta x^{k+1} &= Ea + \rho\mathcal{A}_{adj}X^k - \mathcal{A}_{adj}Z^k + \rho\delta x^k - \rho\mathcal{A}_{adj}\mathcal{A}(x^k) - Ex^k \\ x^{k+1} &= x^k + \frac{1}{\rho\delta}(\mathcal{A}_{adj}(\rho X^k - Z^k - \rho\mathcal{A}(x^k)) + E(a - x^k)) \end{aligned}$$

The main motivation for introducing the proximal terms is to weaken the imposed convergence conditions rather than for the sake of cancellation. [4]

Stopping criteria

The algorithm will terminate if $\|r_p\|_F \leq \epsilon_p$ and $\|r_d\|_2 \leq \epsilon_d$, where

- $r_p = \mathcal{A}(x) - X$ is the primal residual;

[4] For details to see: X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on Bregman iteration", J. Sci. Comput., 46 (2011), pp. 20–46.

- $r_d = \rho\mathcal{A}_{adj}(X_{prev} - X)$ is the dual residual, where X_{prev} is the value of X in the previous iteration; in fact the dual feasibility for the problem is $0 \in E(x^* - a) + \rho\mathcal{A}_{adj}Z^*$ (where x^* and Z^* are the optimal solutions), so if x^{k+1} minimizes $L_\rho(x, X^k, Z^k)$ then

$$\begin{aligned}
&0 \in E(x^{k+1} - a) + \rho\mathcal{A}_{adj}\mathcal{A}(x^{k+1}) - \rho\mathcal{A}_{adj}X^k + \mathcal{A}_{adj}Z^k \\
&0 \in E(x^{k+1} - a) + \mathcal{A}_{adj}(\rho\mathcal{A}(x^{k+1}) - \rho X^k + Z^k) \\
&0 \in E(x^{k+1} - a) + \mathcal{A}_{adj}(\rho\mathcal{A}(x^{k+1}) - \rho X^{k+1} + \rho X^{k+1} - \rho X^k + Z^k) \\
&0 \in E(x^{k+1} - a) + \mathcal{A}_{adj}(\rho r_p^{k+1} + \rho(X^{k+1} - X^k) + Z^k) \\
&\rho\mathcal{A}_{adj}(X^k - X^{k+1}) \in E(x^{k+1} - a) + \rho\mathcal{A}_{adj}Z^{k+1}
\end{aligned}$$

- $\epsilon_p = \sqrt{pq}\epsilon_{abs} + \epsilon_{rel}\max\{\|\mathcal{A}(x)\|_F, \|X\|_F\}$;
- $\epsilon_d = \sqrt{n}\epsilon_{abs} + \epsilon_{rel}\|\mathcal{A}_{adj}(Z)\|_2$.

Typical values for the relative and absolute tolerances are $\epsilon_{rel} = 10^{-3}$ and $\epsilon_{abs} = 10^{-6}$.

Instead of using a fixed penalty parameter ρ, one can vary ρ to improve the speed of convergence. An example for a such scheme is to adapt ρ at the end of each iteration, as follows:

$$\rho := \begin{cases} \tau\rho & \text{if } \|r_p\|_F > \mu\|r_d\|_2 \\ \frac{\rho}{\tau} & \text{if } \|r_d\|_2 > \mu\|r_p\|_F \\ \rho & \text{otherwise} \end{cases}$$

Typical values are $\mu = 10$ and $\tau = 2$.

3.3.2 Proximal gradient method

We would like to apply the PGM to the nuclear norm optimization, so we check if the assumptions of Chapter 2 are satisfied:

- let $f(x) := \frac{1}{2}(x - a)^T E(x - a)$. It is a strongly convex function if and only if E is positive definite. However, in our numerical case $E = I$, so it is positive definite. Therefore, f is strongly convex with parameter $\lambda_{min}(E)$, the smallest eigenvalue. The function is also differentiable and with Lipschitz continuous gradient, in fact

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\| &= \|E(x - a) - E(y - a)\| = \|E(x - a - y + a)\| \\
&= \|E(x - y)\| \le \|E\|\|x - y\|
\end{aligned}$$

Now E is positive definite , then $\|E\| > 0$.

- let $g(x) := \|\mathcal{A}(x)\|_*$, it is continuous and a proper convex function, in fact

$$\|\delta\mathcal{A}(x) + (1-\delta)\mathcal{A}(y)\|_* \leq \delta\|\mathcal{A}(x)\|_* + (1-\delta)\|\mathcal{A}(y)\|_*$$

for all $\delta \in [0,1]$, where we used the triangle inequality of norms.

If we apply directly the PGM, we find

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}}\left\{\|\mathcal{A}(x)\|_* + \frac{1}{\lambda}\|x - x^k + \lambda E(x^k - a)\|_F^2\right\} \\ &= prox^F_{\lambda\|\mathcal{A}(\cdot)\|_*}(x^k - \lambda E(x^k - a)) \end{aligned}$$

with $\lambda > 0$ a parameter. However, we are not able to compute $prox^F_{\|\mathcal{A}(\cdot)\|_*}(\cdot)$. An idea can be to write the dual problem and to apply the proximal gradient method to this latter one.

At the end of Chapter 2 we saw that given the problem $\min_x\{f(x) + g(Ax)\}$, its dual is $\min_u\{f^*(-A_{adj}u) + g^*(u)\}$, where f^* and g^* indicated the conjugate convex of f and g rispectively. Then we compute the conjugate functions:

- $f(x) = \frac{1}{2}(x-a)^T E(x-a)$ is a quadratic function with E a positive semidefinite matrix. We remember that

$$\begin{aligned} &f^*(v) = \sup_x\{v^T x - f(x)\} \\ &f^*(v) = \sup_x\{v^T x - \frac{1}{2}(x-a)^T E(x-a)\} \\ &v - E(x-a) = 0 \quad\Longrightarrow\quad x = E^{-1}v + a \\ &f^*(v) = \frac{1}{2}v^T E^{-1}v + v^T a \end{aligned}$$

So in our case:

$$f^*(-A_{adj}(u)) = \frac{1}{2}(A_{adj}(u))^T E^{-1}A_{adj}(u) - (A_{adj}(u))^T a$$

with $u \in \mathbb{R}^{p\times q}$ and $\mathcal{A}_{adj} : \mathbb{R}^{p\times q} \to \mathbb{R}^n$ is the adjoint map of $\mathcal{A}$.

- $g(Ax) = \|\mathcal{A}(x)\|_*$ and we want to compute $g^*(u)$.

  Lemma 12. *The conjugate convex of a norm is the indicator of unit ball for dual norm.*

  Proof. Let $b(\cdot) := \|\cdot\|$ the generic norm. We recall that the dual norm for a generic norm is $\|y\|^D = \sup_{\|x\|\leq 1} x^T y$. To evaluate $b^*(y) = \sup_x (y^T x - \|x\|)$, we distinguish two cases:

  - if $\|y\|^D \leq 1 \implies y^T x \leq \|x\| \; \forall x \implies$ equality holds if $x = 0 \implies b^*(y) = 0$;
  - if $\|y\|^D > 1 \implies \exists x$ with $\|x\| \leq 1$ and $x^T y > 1 \implies b^*(y) \geq y^T(tx) - \|tx\| = t(y^T x - \|x\|) \xrightarrow[t\to\infty]{} +\infty$

  Then we have

  $$b^*(y) = \begin{cases} 0 & \text{if } \|y\|^D \leq 1 \\ +\infty & \text{if } \|y\|^D > 1 \end{cases}$$

  □

  In our case the dual norm of the nuclear norm is the spectral norm, which returns the largest singular value of the matrix. Therefore

  $$g^*(u) = \begin{cases} 0 & \text{if } \sigma_{max}(u) \leq 1 \\ +\infty & \text{if } \sigma_{max}(u) > 1 \end{cases}$$

  where $\sigma_{max}(u)$ is the largest singular value of u.

In this way the dual problem is

$$\min_u \{f^*(-A_{adj}(u)) + g^*(u)\}$$

$$\min_{u\in\mathbb{R}^{p\times q}} \left\{ \frac{1}{2}(A_{adj}(u))^T E^{-1} A_{adj}(u) - (A_{adj}(u))^T a + g^*(u) \right\} \tag{3.6}$$

To find the solution of 3.6, we use PGM. We saw in Chapter 1 (Example 1) that the proximal operator of an indicator function of a convex set B is the projection onto B. So $prox_{g^*} = \Pi_B$, where $B = \{u : \|u\|_{spectral} \leq 1\}$. Hence we have:

$$\begin{aligned}u^{k+1} &= prox^F_{\lambda g^*}(u^k + \lambda \mathcal{A} \nabla f^*(-\mathcal{A}_{adj}(u^k)) \\ &= prox^F_{\lambda g^*}(u^k - \lambda \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a)) \\ &= \Pi_B(u^k - \lambda \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a))\end{aligned}$$

where u^k indicates the k-th iteration.
Now we will see how to compute the projections onto spectral norm:

- we apply the SVD on $u^k - \lambda \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a) \to U\Sigma V^T$, whre Σ is a diagonal matrix with in the diagonal the singular values of $u^k - \lambda \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a)$;
- we extract the vector of the singular values from $\Sigma = diag(\sigma_i)$;
- we compute the projectorr of the extracted vector using $\|\cdot\|_\infty$

$$\begin{aligned}\hat{\sigma_i} = \Pi_{\|\cdot\|_\infty \leq 1}(\sigma_i) &= \begin{cases} 1 & \text{if } \sigma_i \geq 1 \\ \sigma_1 & \text{if } -1 \leq \sigma_i \leq 1 \\ -1 & \text{if } \sigma_i \leq -1 \end{cases} \\ &= \min\{1, \sigma_i\}\end{aligned}$$

  where the last equality is true because the singular values are all nonnegative;
- we return to the proximal operator of the matrix norm

$$\begin{aligned}u^{k+1} &= \Pi_B(u^k - \lambda \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a)) \\ &= U diag(\hat{\sigma}_i)V^T \\ &= \sum_{i}^{\min\{p,q\}} \min\{1, \sigma_i\} u_i v_i^T\end{aligned}$$

Therefore, the solution of the dual problem can be converted back to the one of the original problem through

$$x^k = \nabla f^*(-\mathcal{A}_{adj} u^k) = -E^{-1}\mathcal{A}_{adj}(u^k) + a$$

Then the algorithm is

```
initialize  $u^0$
for  $k = 0, 1, ..$  until stopping conditions are satisfied
update  $x^k = -E^{-1}\mathcal{A}_{adj}(u^k) + a;$
        $u^{k+1} = \Pi_B(u^k + \lambda \mathcal{A} x^k) = \sum_{i}^{\min\{p,q\}} \min\{1, \sigma_i\} u_i v_i^T$
```

Choice of the parameter λ

Following Chapter 2, if ∇f^* is Lipschitz with constant L then we can choose a fixed step size $\lambda \in (0, 1/L]$. Now, as stated by Observation 2 if f is strongly convex with parameter μ, f^* has a Lipschitz gradient $\frac{\|\mathcal{A}\|^2}{\mu}$, then we have

$$L = \frac{\|\mathcal{A}\|^2}{\mu} = \frac{\sigma_{max}(\mathcal{A})^2}{\lambda_{min}(E)}$$

where $\sigma_{max}(A)$ is the largest singular value of $\mathcal{A}$. Therefore, we can choose $\lambda = \frac{\lambda_{min}(E)}{\sigma_{max}(\mathcal{A})^2}$.

In an equivalent way, we can findd the parameter by a line search as we saw in Paragraph 2.1.

Stopping criteria

The function $f(x) + g(\mathcal{A}(x))$ is a convex function because it is the sum of two convex functions (a quadratic function and a norm). Therefore, the minimum is unique and if there exist two minimum x and y we have that $x - y = 0$. So, it can be used as primal arrest criteria for our algorithm:

$$r_p := x^{k-1} - x^k$$

Then if $\tilde{u}$ is optimal solution of the dual problem, we have that it is a fixed point of the proximal gradient method, i.e.

$$\tilde{u} = \Pi_B(\tilde{u} + \lambda \mathcal{A} \tilde{x})$$

where $\tilde{x}$ is the optimal solution of the primal problem.

So, the algorithm stops if

$$\|r_p\|_2 := \|x^k - x^{k-1}\|_2 \leq \epsilon_p$$
$$\|r_d\|_F := \|u^k - u^{k-1}\|_F \leq \epsilon_d$$

where ϵ_d and ϵ_p are tolerances. Typical values are 10^{-4}.

3.3.3 Fast proximal gradient method

Now we apply the FPGM to our dual problem, so the algorithm becomes:

```
initialize  $u^0 = z^1, \quad t_1 = 1$
for  $k = 0, 1, ..$  until stopping conditions are satisfied
update  $x^k = \nabla f^*(-\mathcal{A}_{adj} u^k) = -E^{-1}\mathcal{A}_{adj}(u^k) + a$
        $u^{k+1} = \Pi_B(z^{k+1} - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(z^{k+1}) - a))$
        $t_{k+2} = \frac{1 + \sqrt{1 + 4{t_k}^2}}{2}$
        $z^{k+2} = u^{k+1} + \frac{t_k - 1}{t_{k+1}}(u^{k+1} - u^k)$
```

In the update of u^{k+1} we use the same strategy that we have seen in the PGM, i.e. we compute a decomposition in singular values of the matrix $z^k - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(z^k) - a)$.

We can use the same step size and the same stopping criterion of the proximal gradient method.

Alternative choice for t_k

If we analyze the proof of the theorem about the convergence of FPGM, we notice that the result is correct if $\{t_k\}_k$ is a sequence satisfying the following properties for any $k \geq 0$:

- $t_k \geq \frac{k+1}{2}$

- $t_k^2 \geq t_{k+1}^2 - t_{k+1}$

So every choice of t_k which satisfy these properties is valid. For example, Nesterov, in his work in which he presented this algorithm, proposed to use $t_k = \frac{k+1}{2}$, but other values go well, for example $t_k = \frac{k+2}{2}$ or $t_{k+1} = \frac{1+\sqrt{1+4{t_k}^2}}{2}$, like above.

3.3.4 Proximal Newton method

We consider the problem

$$\min_x \{f(x) + g(x)\}$$

where

- $f(x) = \frac{1}{2}(x-a)^T E(x-a)$, is a proper convex, continuosly, differentiable function and its gradient is Lipschitz continuos. Moreover

$$\nabla f(x) = E(x-a) \quad \nabla^2 f(x) = E.$$

- $g(x) = \|\mathcal{A}(x)\|_*$ is a proper convex but not differentiable function.

Following Chapter 2, we apply proximal Newton method: starting with x^0. For $k = 1, 2, ...$ until stopping conditions are satisfied:

$$d^k = \operatorname*{argmin}_z \big\{ \nabla f(x^{k-1})^T (z - x^{k-1}) + \frac{1}{2}(z - x^{k-1})^T H^{k-1} (z - x^{k-1}) + g(z) \big\}$$
$$x^k = x^{k-1} + t^k (d^k - x^{k-1})$$

where H^{k-1} is an approximationof the Hessian of $f(x^{k-1})$. In our case the Hessian is easy to compute: it is the matrix E. We saw that in an equivalent formulation

$$\begin{aligned} d^k &= prox_g^E(x^{k-1} - E^{-1}\nabla f(x^{k-1})) \\ &= prox_g^E(x^{k-1} - E^{-1}E(x^{k-1} - a)) \\ &= prox_g^E(x^{k-1} - x^{k-1} + a) \\ &= prox_g^E(a) \\ &= \operatorname*{argmin}_z \left\{ g(z) + \frac{1}{2}(z-a)^T E(z-a) \right\} \\ &= \operatorname*{argmin}_z \{g(z) + f(z)\} \end{aligned}$$

We notice that we have the initial problem, therefore it is not useful to apply the proximal Newton method to this type of problem. In the inner subproblem, we obtain the initial problem because to compute the proximal Newton method direction $d^k - x^{k-1}$, we minimize a quadratic approximation in f using the hessian, plus original g. However, f is already a quadratic function, so we meet again to minimize the original problem.

An idea can be written the dual problem and to apply the proximal Newton method to this latter one, as in the PGM. In fact, given the problem

$$\min\{f(x) + g(Ax)\}$$

where

- $f(x) := \frac{1}{2}(x-a)^T E(x-a)$;
- $g(Ax) := \left\|\mathcal{A}(x)\right\|_*$,

the Lagrangian of the problem is

$$\mathcal{L}(x, y, u) = f(x) + g(y) + \langle u, \mathcal{A}(x) - y\rangle$$

Increasing and minimizing the Lagrangian, we find the dual problem:

$$\min_u\{f^*(-A_{adj}(u)) + g^*(u)\}$$

$$\min_{u\in\mathbb{R}^{p\times q}}\left\{\frac{1}{2}(A_{adj}(u))^T E^{-1} A_{adj}(u) - (A_{adj}(u))^T a + g^*(u)\right\}$$

We remember that g^* is the conjugate convex of g and in this case it is a indicator function.

Now, if $\tilde{u}$ is an optimal solution of the dual problem and $\tilde{x}$ is an optimal solution of the primal problem, using the optimality conditions, we have

$$0 \in \nabla f(\tilde{x}) + \mathcal{A}_{adj}(\tilde{u}) = E(\tilde{x} - a) + \mathcal{A}_{adj}(\tilde{u})$$

So, if we solve the dual problem, any solution can be converted back to the one of the original problem through

$$\tilde{x} = -E^{-1}\mathcal{A}_{adj}(\tilde{u}) + a = \nabla f^*(-\mathcal{A}_{adj}(\tilde{u})).$$

Then, to solve the dual problem we use the proximal quasi-Newton method, because to compute the Hessian of $f^*(-\mathcal{A}_{adj}(\tilde{u}))$ is difficult. The algorithm is :

```
**given** $u^0$, $\nabla^2 f^*(-\mathcal{A}_{adj}(u^0)) = H^0$, $\eta_0 = 0.5$
**repeat** for $k = 1, 2, ...$, until stopping conditions are satisfied
    1. **update** $x^{k-1} = -E^{-1}\mathcal{A}_{adj}(u^{k-1}) + a$
    2. solve $d^k = \text{argmin}_d \nabla f^*(-\mathcal{A}_{adj}(u^{k-1}))^T d + \frac{1}{2} d^T H^{k-1} d + g^*(u^{k-1}+d)$
    3. using $\eta_{k-1}$ verify if the inexact solution of the subproblem is valid
    4. select $t^k$ with backtracking line search
    5. **update** $u^k = u^{k-1} + t^k d^k$
    6. **update** $\eta_k$
    7. **update** $H^k$
```

Search direction

If we use the scaled proximal mapping, the search direction is

$$d^k = prox_{g^*}^{H^{k-1}}(u^{k-1} - (H^{k-1})^{-1}\nabla f^*(-\mathcal{A}_{adj}(u^{k-1})))$$

When $H^{k-1} = I$, this is a projection of $u^{k-1} - \nabla f^*(-\mathcal{A}_{adj}(u^{k-1}))$ onto the set described by the indicator function g^*. However, in general, it is not a projection, but it is more complicated. Then, to solve the subproblem we use the equivalent definition:

$$\begin{aligned} d^k = \underset{z \in \mathbb{R}^{p \times q}}{\text{argmin}} \Big\{ & (\nabla f^*(-\mathcal{A}_{adj}(u^{k-1})))^T (z - u^{k-1}) + \frac{1}{2}(z - u^{k-1})^T H^{k-1}(z - u^{k-1}) \\ & + g^*(z) \Big\} \\ = \underset{z \in \mathbb{R}^{p \times q}}{\text{argmin}} \Big\{ & (\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a)^T (z - u^{k-1}) + \frac{1}{2}(z - u^{k-1})^T H^{k-1}(z - u^{k-1}) \\ & + g^*(z) \Big\} \end{aligned}$$

In practice, it is expensive to solve this subproblem accurately, so we perform this inner minimization inexactly, i.e. we use an inexact proximal Newton method. We compute the solution using the FPGM. Let

- $h(z) := (\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1})-a)^T(z-u^{k-1})+\frac{1}{2}(z-u^{k-1})^T H^{k-1}(z-u^{k-1})$, it is a quadratic function and it is strongly convex if and only if H^{k-1} is positive definite. However, H^{k-1} is the approximation of the Hessian of a convex function, so it is constructed positive definite. Then, h is differentiable and its gradient is Lipschitz continuos. In fact

$$\nabla h(z) = (\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a) + H^{k-1}(z - u^{k-1})$$

$$\|\nabla h(z_1) - \nabla h(z_2)\| = \|H^{k-1}(z_1 - z_2)\| \leq \|H^{k-1}\| \|z_1 - z_2\|$$

  $\|H^{k-1}\| > 0$ because the matrix is positive definite.

- $s(z) = g^*(z) = \begin{cases} 0 & \text{if } \sigma_{max}(z) \leq 1 \\ +\infty & \text{if } \sigma_{max}(z) > 1 \end{cases}$. The indicator function is convex and it has an efficient proximal mapping.

Therefore, we have to solve the following problem:

$$\underset{z}{\operatorname{argmin}}\{h(z) + s(z)\}$$

We can use the FPGM to find the search direction:

```
**given** $b^1 = z^0$, $c_1 = 1$
**for** $i = 1, 2, ...$
    $z^i = prox_{\lambda s}(b^i - \lambda\nabla h(b^i))$
        $= \Pi_B(b^i - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a) - \lambda H^{k-1}(b^i - u^{k-1}))$
        $= \sum_{j=1}^{min\{p,q\}} min\{1, \sigma_j\} u_j v_j^T$
    $c_{i+1} = \frac{1+\sqrt{1+4c_i^2}}{2}$
    $b^{i+1} = z^i + \frac{c_1 - 1}{c_{i+1}}(z^i - z^{i-1})$
```

where σ_j are the singular values of the matrix $b^i - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a) - \lambda H^{k-1}(b^i - u^{k-1}) = U diag(\sigma_j) V^T$.

Now a matrix z^i is accepted as an approximate proximal Newton step if it satisfies the following properties, as we explained in Paragraph 2.4.2:

- let

  $$\begin{aligned}\varphi_{k-1}(z) &:= f^*(-\mathcal{A}_{adj}(u^{k-1})) + (\nabla f^*(-\mathcal{A}_{adj}(u^{k-1})))^T(z-u^{k-1}) + \\ &+ \frac{1}{2}(z-u^{k-1})^T H^{k-1}(z-u^{k-1}) + g^*(z) \\ &= f^*(-\mathcal{A}_{adj}(u^{k-1})) + h(z) + s(z)\end{aligned}$$

  then

  $$\varphi_{k-1}(z^i) \leq \varphi_{k-1}(u^{k-1})$$

- $\|G_{\varphi_{k-1}}(z^i)\| \leq \eta_{k-1}\|G_\varphi(x^{k-1})\|$

  where

  - $\varphi(z) = f^*(-A_{adj}(z)) + g^*(z)$;
  - $G_\varphi(z^i) = z^i - prox_{g^*}(z^i - \lambda\nabla f^*(-A_{adj}(z^i)))$
    $= z^i - \Pi_B(z^i - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(z^i) - a))$
    $= z^i - \sum_i \min\{1, \sigma_i\} u_i v_i^T$

    where σ_i are the singular values of the matrix $z^i - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(z^i) - a) = Udiag(\sigma_i)V^T$;

  - $G_{\varphi_{k-1}}(z^i) = z^i - prox_{g^*}(z^i - \lambda\nabla f^*(-A_{adj}(u^{k-1})) - \lambda H^{k-1}(z^i - u^{k-1}))$
    $= z^i - \Pi_B(z^i - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a) - \lambda H^{k-1}(z^i - u^{k-1}))$
    $= z^i - \sum_i \min\{1, \sigma_i\} u_i v_i^T$

    where σ_i are the singular values of the matrix $z^i - \lambda\mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a) - \frac{1}{2}H^{k-1}(z^i - u^{k-1}) = Udiag(\sigma_i)V^T$;

So, if we suppose that the algorithm terminates at the p-th iteration, the search direction is

$$d^k = z^p \implies \Delta_k := d^k - u^{k-1}\text{is the search direction}$$

Selection of $\mathbf{t^k}$

We select t^k with backtracking line search.Let $t^k = t = 1$,

$$
\begin{aligned}
&\varphi(u^{k-1} + t\Delta_k) - \varphi(u^{k-1}) = f^*(-\mathcal{A}_{adj}(u^{k-1} + t\Delta_k)) - f^*(-\mathcal{A}_{adj}(u^{k-1}))+\\
&+ g^*(u^{k-1} + t\Delta_k) - g^*(u^{k-1})\\
&= f^*(-\mathcal{A}_{adj}(u^{k-1} + t\Delta_k)) - f^*(-\mathcal{A}_{adj}(u^{k-1})) + g^*(u^{k-1} + t\Delta_k - tu^{k-1} + tu^{k-1}) - g^*(u^{k-1})\\
&= f^*(-\mathcal{A}_{adj}(u^{k-1} + t\Delta_k)) - f^*(-\mathcal{A}_{adj}(u^{k-1})) + g^*((1-t)u^{k-1} + t(\Delta_k + u^{k-1})) - g^*(u^{k-1})\\
&\leq f^*(-\mathcal{A}_{adj}(u^{k-1} + t\Delta_k)) - f^*(-\mathcal{A}_{adj}(u^{k-1})) + tg^*(\Delta_k + u^{k-1}) - tg^*(u^{k-1})\\
&\approx (\nabla f^*(-\mathcal{A}_{adj}(u^{k-1}))^T \mathcal{A}_{adj}(-t\Delta_k) + tg^*(\Delta_k + u^{k-1}) - tg^*(u^{k-1})\\
&= t[(\nabla f^*(-\mathcal{A}_{adj}(u^{k-1}))^T \mathcal{A}_{adj}(-\Delta_k) + g^*(u^{k-1} + \Delta_k) - g^*(u^{k-1})]
\end{aligned}
$$

where we used the fact that g^* is a convex function. So **while**

$$\varphi(u^{k-1}+t\Delta_k)-\varphi(u^{k-1})-t[(-E^{-1}\mathcal{A}_{adj}(u^{k-1})+a)^T\mathcal{A}_{adj}(-\Delta_k)+g^*(u^{k-1}+\Delta_k)-g^*(u^{k-1})] > 0$$

then

$$t = \frac{t}{2}$$

Updating of $\mathbf{u^k}$

We update the dual solution:

$$u^k = u^{k-1} + t^k\Delta_k$$

Updating of $\boldsymbol{\eta_k}$

We update the forcing term

$$\eta_k = \min\left\{\frac{m}{2}, \frac{\|G_{\varphi_{\tilde{k-1}}/M}(u^k) - G_{\varphi/M}(u^k)\|}{\|G_{\varphi/M}(u^{k-1})\|}\right\}$$

where $mI \preceq H^k \preceq MI$.

Updating of H^k

To update the approximation of the Hessian we do not use the BFGS formula, but its dual version, i.e. the DFP formula because we are computing the approximation of the Hessian of the dual of f. In the Paragraph 2.4.1 we saw that the DFP formula is

$$B^k = B^{k-1} + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} - \frac{B^{k-1}y^{k-1}(y^{k-1})^T B^{k-1}}{(y^{k-1})^T B^{k-1} y^{k-1}}$$

where

- B^k is the approximation of the inverse of the Hessian,
- $s_{k-1} := u^k - u^{k-1}$,
- $$\begin{aligned} y^{k-1} &:= \nabla f^*(-\mathcal{A}_{adj}(u^k)) - \nabla f^*(-\mathcal{A}_{adj}(u^{k-1})) \\ &= \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a) - \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^{k-1}) - a) \\ &= \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k) - a + E^{-1}\mathcal{A}_{adj}(u^{k-1}) + a) \\ &= \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k - u^{k-1})) \end{aligned}$$

To achieve the approximation of the Hessian we apply the Sherman-Morrison-Woodbury formula (to see Appendix A for details):

$$H^k = \left(I - \frac{y^{k-1}s_{k-1}^T}{(y^{k-1})^T s_{k-1}}\right) H^{k-1} \left(I - \frac{s_{k-1}(y^{k-1})^T}{(y^{k-1})^T s_{k-1}}\right) + \frac{y^{k-1}(y^{k-1})^T}{(y^{k-1})^T s_{k-1}}$$

We verify that $(y^{k-1})^T s_{k-1}$ is a scalar, infact H^k, H^{k-1}, $y^{k-1}s_{k-1}^T$, $s_{k-1}(y^{k-1})^T$, $y^{k-1}(y^{k-1})^T$ are all matrices $pq \times pq$

$$\begin{aligned} (y^{k-1})^T s_{k-1} &= \langle s_{k-1}, y^{k-1} \rangle = \langle u^k - u^{k-1}, \mathcal{A}(E^{-1}\mathcal{A}_{adj}(u^k - u^{k-1})) \rangle \\ &= \langle \mathcal{A}_{adj}(u^k - u^{k-1}), E^{-1}\mathcal{A}_{adj}(u^k - u^{k-1}) \rangle \\ &= \underbrace{(E^{-1}\mathcal{A}_{adj}(u^k - u^{k-1}))^T}_{1\times n} \underbrace{\mathcal{A}_{adj}(u^k - u^{k-1})}_{n\times 1} \\ &= \text{scalar} \end{aligned}$$

Stopping criteria

If $\tilde{x}$ and $\tilde{u}$ are the optimal solutions of the primal and dual problem respectively, we have:

- minimum of the primal problem is unique because the function is convex,
- gradient of φ at point $\tilde{u}$ is zero

However, we are not able to compute the gradient of φ because the function g^* is non-smooth. Therefore, we use the definition of generalized gradient previous presented. In fact

$$G_\varphi(u) = u - prox_{g^*}(u - \lambda\nabla f^*(\mathcal{A}_{adj}(u)))$$

We have

$$G_\varphi(\tilde{u}) = 0 \qquad \Longleftrightarrow \qquad \tilde{u} = prox_{g^*}(\tilde{u} - \lambda\nabla f^*(\mathcal{A}_{adj}(\tilde{u})))$$

Remembering the interpretation of the proximal gradient method as fixed point iteration, we have that $\tilde{u}$ is a fixed point of the algorithm $\Longleftrightarrow$ $\tilde{u}$ is a minimizes of φ.

Then, we can use them as stopping criterion, i.e. the algorithm stops if

$$\begin{aligned} \|r_p\|_2 &:= \|x^k - x^{k-1}\|_2 \le \epsilon_p \\ \|r_d\|_F &:= \|G_\varphi(u^k)\|_F \le \epsilon_d \end{aligned}$$

where ϵ_d and ϵ_p are tolerances.

Chapter 4

Numerical experiments

In this section, we report results on numerical experiments in Python 3. We consider two cases: in the first one, the measured inputs and outputs are complete. In the second case, instead, a percentage of outputs is removed.

Once the optimized output is calculated through a proximal algorithm method, we reconstruct the system matrices using a subspace method. We do not use instrumental variables and, given n the estimated order, to estimate the observability matrix O_n we compute the singular value decomposition of $G = Y_{o,n,N}\Pi_{0,n,N} = O_n X_{0,1,N}\Pi_{0,r,N}$.

$$G = \begin{bmatrix} P_n & P_e \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_e \end{bmatrix} \begin{bmatrix} Q_n & Q_e \end{bmatrix},$$

where with P_n we indicate the first n columns of the matrix P. In this way, we have that $O_n \approx P_n\Sigma_n$.

We can consider two ways to estimate the order of the model:

- to use a threshold on the singular values, i.e. we count the number of singular values which are greater or equal than 10^{-3} multiplied by the maximum singular value. We set a maximum model order equal to 10;

- to use the parsimony criterion AIC (Akaike's Information Criterion)

$$\log\left(V \cdot \left(1 + \frac{n \cdot 2}{N}\right)\right) \approx \log(V) + \frac{n \cdot 2}{N} \tag{4.1}$$

  i.e. we search an order n such that the function 4.1 is minimum, where

– V is the loss function, $V = \frac{1}{N}\sum_{k=1}^{N}(y(k) - y_{meas}(k))^2$;

– $y_{meas}(k)$ is the measured output;

– n is the model order

– N is the number of observations used to estimate the optimized output

To compare the quality of different models and algorithms, we use the validation fit measure. It is defined in percentage as

$$fit = 100\left(1 - \frac{\|y - y_{est}\|}{\|y - mean(y)\|}\right)$$

for a single output sequence, where y is the validation data output and y_{est} is the estimated output from the model. For system with multiple outputs, we report the average of the fit. Another criterion that we use is the relative error:

$$error = \frac{\|y - y_{est}\|}{\|y\|}$$

The nuclear norm optimization problems are solved using the previous proximal algorithms: ADMM, PGM, FPGM and PNM. The maximum number of iterations is set to 200.

4.1 Complete inputs and outputs

In this first set of experiments we solve the nuclear norm optimization problem 3.4, i.e. we computed a modified output sequence. In all the experiments the parameter r (the number of rows of the Hankel matrices) is equal to 15.

4.1.1 Examples from the DaISy collection

In this section, we consider four examples from the DaISy collection [11]. Since there is only one input/output sequence for each dataset, we break up the data sequence in two sections: the first N_I data points are used for the identification, the next N_V for validation.

The four datasets are:

- 96-006, hair dryer. It describes a dryer and there are one input and one output. The input is the voltage over the heating device, while the output is the air temperature which is measured by a thermocouple. We choose $N_I = 300$ and $N_V = 700$.

- 96-007, CD player arm. It is data from the mechanical construction of a CD player arm. The inputs are the forces of the mechanical actuators, while the outputs are related to the tracking accuracy of the arm. The data is measured in a closed loop, and then through a two-step procedure converted to open loop equivalent data, so there are two input and two outputs. We choose $N_I = 100$ and $N_V = 400$.

- 96-009, robot arm. Data come from a flexible robot arm. The arm is installed on an electrical motor. The authors have modeled the transfer function from the measured reaction torque of the structure on the ground to the acceleration of the flexible arm. The applied input is a periodic sine sweep. Then, the input is the reaction torque of the structure and the output the acceleration of the flexible arm. We choose $N_I = 300$ and $N_V = 700$.

- 96-011, thermic res wall. There are two inputs and one output and it describes the heat flow density through a two-layer wall (brick and insulation layer). The inputs are the internal and external temperature of the wall, while the output is the heat flow density through the wall. We choose $N_I = 400$ and $N_V = 1000$.

In the first experiment, we consider the hair dryer with $N_I = 300$ and $N_V = 700$. Firstly we study only the ADMM. The figure 4.1 shows the singular values of matrix $Y\Pi$ constructed from the optimized outputs and the matrix $Y_{meas}\Pi$ constructed from the measured outputs. Since we want the new outputs to have a minimum nuclear norm, but still close to the measured outputs, it is right that the singular values obtained are below those obtained from the original matrix.

We now study the behavior of the fit as the number of iterations increases. If we set a maximum number of iterations equal to 1000, it seems that the fit reaches a maximum and then stabilizes. However, if we go to zoom we see oscillations and, from the iteration 500 onwards, there is a slow decrease in the fit, figure 4.2.

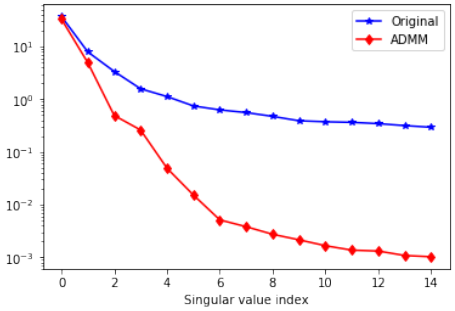


Figure 4.1: singular values from original and optimized outputs

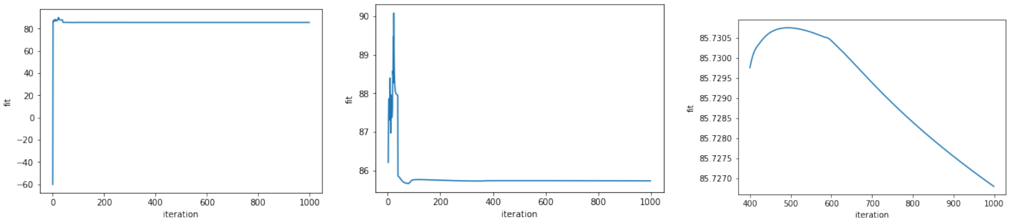


Figure 4.2: Behavior of the fit

We have this behavior because the problem that ADMM is going to solve minimizes, on the one hand, the nuclear norm of $Y\Pi$ and on the other, it tries to have a minimum distance from the measured outputs. These two values influence the estimation of the observability matrix and the order of the model. In fact, a small nuclear norm corresponds to an estimated low order of the model, so there is the risk of underestimating. On the other hand, stopping at high nuclear norm can overestimate the model. Consequently, it is necessary to look for a number of iterations for which the fit (or equivalently the relative error) is maximum (minimum).

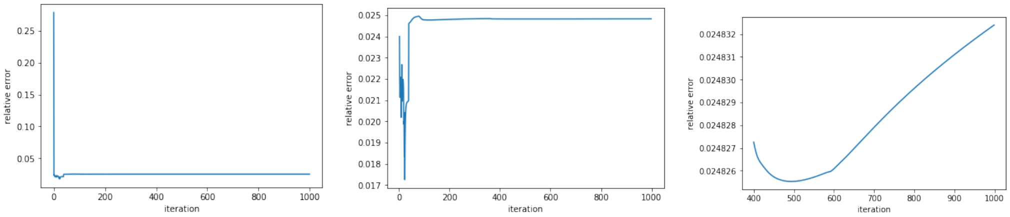


Figure 4.3: Behavior of the relative error

So, in the case of the ADMM we have the best fit at iteration 23. We have fit 90.08 and estimated order 5. If instead, we use the original outputs, we have fit 87.21 and estimated order 10.

In figure 4.4, we can compare the original outputs with the outputs of the identified model. In the figure on the left, to estimate the state-space matrix we used the optimized ouputs from ADMM, instead in the figure on the right we used the measured outputs.

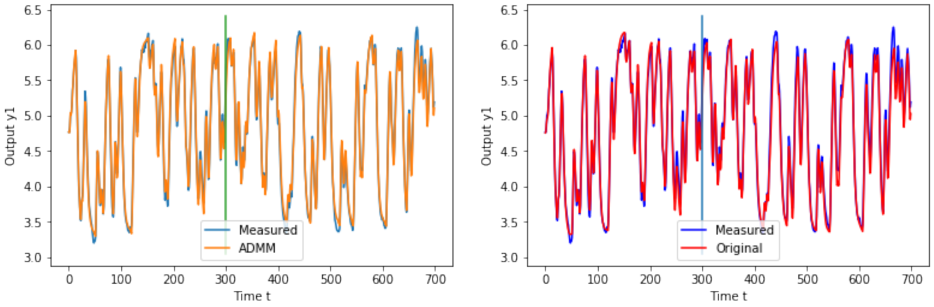


Figure 4.4

Now, we compare the four proximal algorithms. We consider both the cases to estimate the model order. So with $fit1$ we indicate the fit obtained from the estimation of the order from a threshold on the singular values, with $fit2$ the fit obtained with the use of the parsimony criterion AIC.

Dataset	fit1	estimated order	fit2	estimated order with AIC
$96-006$	87.21	10	88.39	6
$96-007$	57.14	10	56.26	1
$96-009$	96.44	7	96.69	5
$96-011$	85.42	10	86.76	9

Table 4.1: Fit and estimated order from the original measured outputs

From Tables 4.1, 4.2, 4.3, 4.4 and 4.5 we note that with PGM, in the case of CD player arm, we get lower fit values than FPGM, in fact from the theory we know that the convergence speed of the latter is equal to $1/k^2$, where k is the number of iterations, while the convergence of PGM has speed $1/k$. Consequently, if we bring the maximum number of iterations to 735, we see

	fit1	order	iter.	fit2	order	iter.
ADMM	75.82	5	98	79.14	5	13
PGM	61.28	10	3	60.85	5	58
FPGM	76.23	5	129	78.32	5	18
PNM	58.23	10	20	77.92	7	11

Table 4.2: Fit and estimated order of the CD player arm (96-007)

	fit1	order	iter.	fit2	order	iter.
ADMM	90.09	6	12	89.90	6	7
PGM	90.10	5	55	90.14	7	3
FPGM	89.46	6	9	90.01	5	90
PNM	88.55	10	2	89.85	4	1

Table 4.3: Fit and estimated order of the hair dryer (96-006)

	fit1	order	iter.	fit2	order	iter.
ADMM	83.21	10	27	81.35	8	17
PGM	66.12	10	18	82.04	5	58
FPGM	82.79	10	112	82.21	9	122
PNM	79.31	10	2	96.69	5	1

Table 4.4: Fit and estimated order of the robot arm (96-009)

	fit1	order	iter.	fit2	order	iter.
ADMM	86.85	6	19	86.85	7	9
PGM	86.86	10	8	86.85	6	26
FPGM	86.84	10	5	86.84	7	9
PNM	86.80	10	8	86.80	6	13

Table 4.5: Fit and estimated order of the thermic res wall (96-011)

that PGM reaches a fit of 75.04 and estimated order 7. In the figure 4.5 we can see the trend of the residue $\|x_{k+1} - x_k\|$, where x_k is the solution to the primal problem, and in the case of FPGM we can see the Nesterov waves.

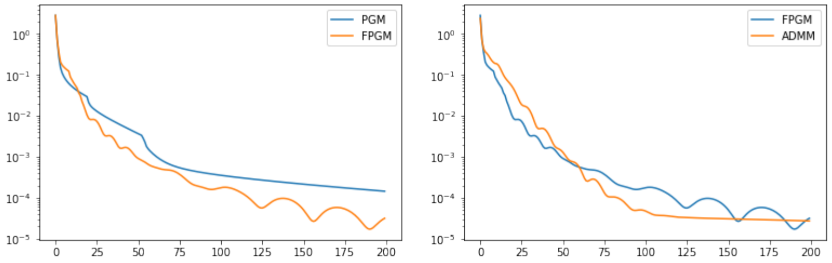


Figure 4.5: (Left) $\|x_{k+1} - x_k\|$ of PGM and FPGM, (right) $\|x_{k+1} - x_k\|$ of ADMM and FPGM

In all cases, FPGM and ADMM obtain similar results, although the number of iterations that FPGM must do is sometimes higher due to the formation of Nesterov waves.

Now we do not want to find the minimum of the problem, as we risk having an excessively small nuclear norm and, therefore, underestimating the order.

As for PNM, since the function is quadratic, in a single iteration we could reach the minimum. It was preferred, then, to use quasi-PNM to have more possibilities of combinations of nuclear norm/distance from the measured output. However the latter method reaches the minimum more quickly, consequently, there are fewer combinations of nuclear norm/distance and, therefore, there is the risk of not reaching an optimal fit. In figure 4.6, we can see the decrease of values of the dual problem.

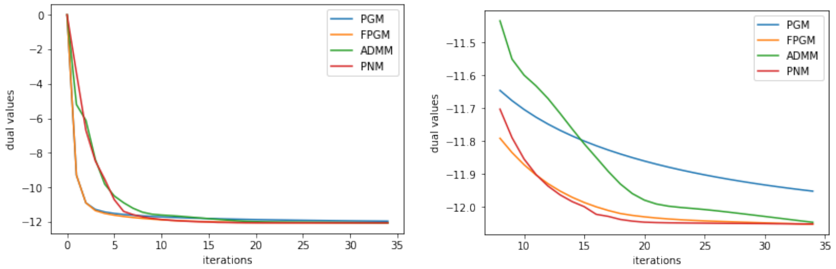


Figure 4.6: Values of dual problem for the hair dryer

In the PNM at each iteration we have to calculate a direction of descent, to do this we solve inexactly the minimum problem using ADMM, PGM or

FPGM. Due to the Nesterov waves, with the last method we must also verify that the direction found is actually a downward direction. In the tables below we can see the results:

	fit	order	iter.
FPGM	58.23	10	20
ADMM	51.99	10	13
PGM	60.38	10	15

	fit	order	iter.
FPGM	88.55	10	2
ADMM	88.71	10	2
PGM	88.52	10	2

Table 4.6: (Left) methods used to find search direction for CD player arm, (right) methods used to find search direction for hair dryer

A negative aspect of the PNM is its high computational burden. In fact, at each iteration it is required to find a direction of descent solving a minimum problem, finding a suitable step along it and estimating the Hessian matrix. As a result, calculation times are longer.

If we consider the two methods to estimate the order of the model, we see a clear reduction of the estimated order in the case of the method based on the parsimony principle AIC. In fact, this second method tends to underestimate the order, rather than overestimate it, but this is better because we avoid problems e.g. the over-fitting. We notice, also, a greater fit.

4.1.2 Example from generated data

In this section, we try to understand why the method for estimating the order based on the AIC parsimony principle works better than the one based on a threshold on singular values.

We consider the following state-space model of order $n_x = 7$ e with the system matrices:

$$A = T \begin{bmatrix} 0.9 & 0 & \dots & & & & 0 \\ 0 & 0.8 & 0 & \dots & & & 0 \\ 0 & 0 & 0.4 & 0 & \dots & & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & \dots & & 0 & 0.2 & 0 & 0 \\ 0 & & \dots & & 0 & 0.1 & 0 \\ 0 & & & & \dots & 0 & 0.001 \end{bmatrix} T$$

with T deriving from the QR factorization of a random matrix,

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We choose the vector u in a deterministic way and add a pseudo-random sequence through the command *randn*, in order to generate an input which excites all the dynamics of the system. We consider $N_I = 100$ and $N_V = 400$.

In the first experiment, we consider 20 pairs of inputs/outputs and suppose that they are measured without noise. To solve the problem we apply the ADMM algorithm and compare the two methods to estimate the order of the model.

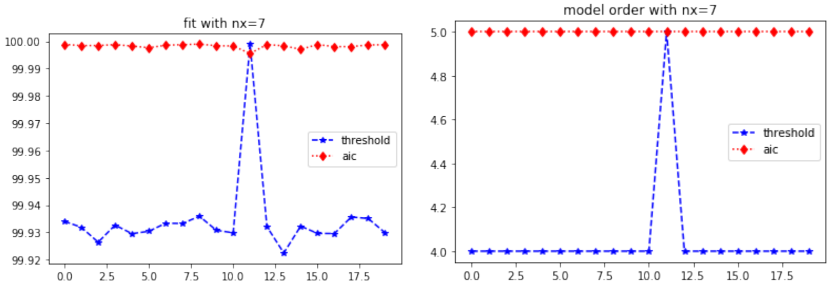


Figure 4.7: Fit and estimated order for 20 pairs of input/output measured without noise

In figure 4.7, we can see that the method which uses a threshold on singular values gives an estimated order 4, while the method which uses the AIC principle returns 5. In both cases, we have a very high fit in all the examples.

Now, let's add noise to the measured data. We note that with the "AIC" method we obtain a greater fit, furthermore, we have an estimated order fixed at 4, while with the "threshold" method the order is always 10, figures 4.8.

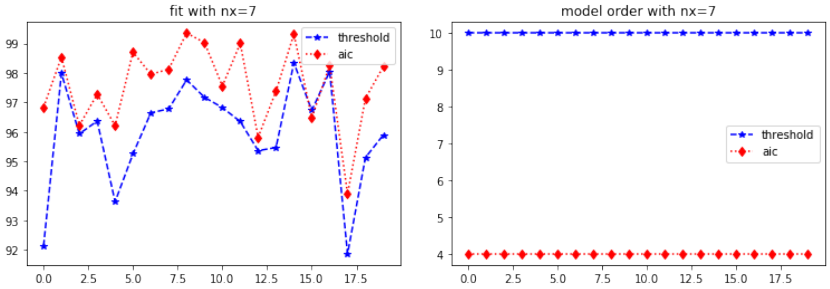


Figure 4.8: Fit and estimated order for 20 pairs of input/output measured with noise

We consider the matrices of the state-space system estimated by the first pair of inputs/outputs, take the remaining inputs, pass them through the estimated system and check if the fit remains stable. From the figure 4.9, we see that in the "AIC" case the fit is more stable than the other one.

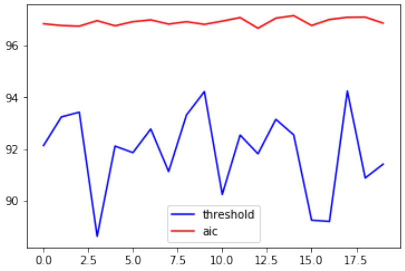


Figure 4.9: Stability of the fit

This happens because the singular values are very sensitive to the added noise, so the "AIC" method behaves better than the method that uses the threshold on singular values. This, then, also explains the behavior of the two methods in the case of DaISy databases, in fact the data are measured with noise, so it makes sense to get better results with "AIC".

4.2 Missing outputs

In this set of experiments, we evaluate the nuclear norm approach for the problem with missing outputs. We consider the data from the DaISy database but remove a percentage of randomly chosen outputs from the identification sequence. In this case with the nuclear norm optimization problem, we reconstruct the missing outputs. We also compare the results obtained by the proximal algorithms with another method: first reconstructing the output by linear interpolation and then using it to estimate the matrices of the system.

From the following tables, it can be seen that both in the case of the CD player arm and of the thermic res wall the nuclear norm optimization approach works well with a high percentage of missing data. An exception is given by the example of the hair dryer, in which already with a 10% of missing data the fit is low. With 20% of missing data, negative fit values are recorded. However, if we draw the graph of the estimated outputs by the matrices of the system, with the output sequence of the validation, we see that the estimated outputs have a trend similar to true outputs, but translated along the y-axis, figure 4.10.

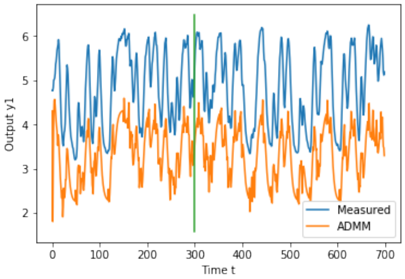


Figure 4.10: Estimated outputs for hair dryer with 30% of missing outputs.

This derives from the fact that the nuclear norm problem requires that the estimated outputs are close to the measured outputs. Now, in this case, the measured outputs have the 70% of the values comprised between 4 and 6 and the remainder equal to 0. Consequently, when the proximal algorithm reconstructs the outputs it takes into account the presence of such 0, therefore even if he finds the trend, the optimized outputs are translated and closer to the x-axis.

10%		fit	order
	interpol.	24.94	10
	ADMM	33.43	10
	PGM	25.54	10
	FPGM	26.54	10
	PNM	25.65	10

Table 4.7: Missing outputs for hair dryer. The order is estimated with a threshold on singular values.

10%		fit	order
	interpol.	32.65	8
	ADMM	34.70	3
	PGM	12.24	7
	FPGM	12.24	7
	PNM	16.07	4

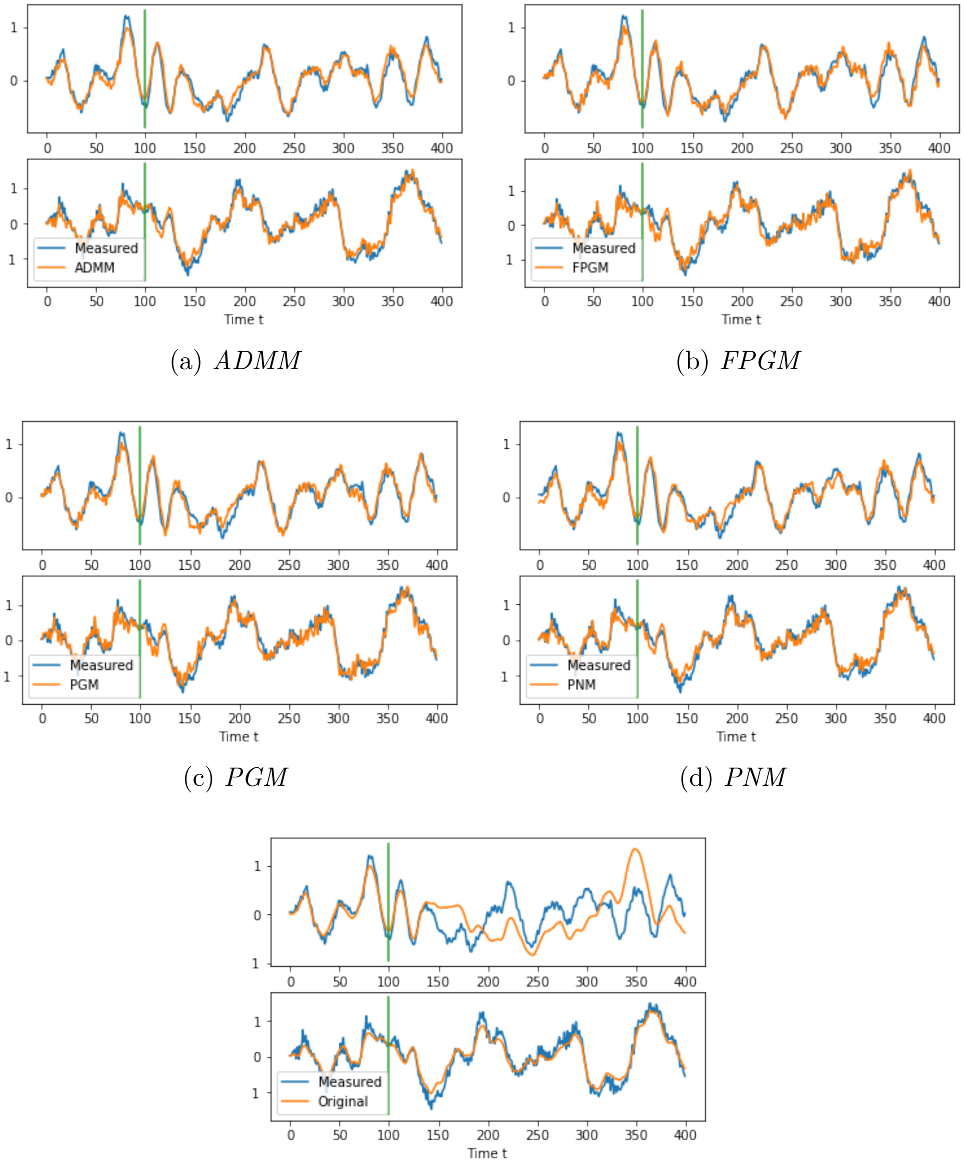
Table 4.8: Missing outputs for hair dryer. The order is estimated with the AIC principle.

10%		fit	order	fit	order	fit	order
	interpol.	15.14	10	75.20	10	88.17	10
	ADMM	71.58	6	78.88	5	75.29	10
	PGM	70.83	10	75.69	10	69.47	10
	FPGM	72.65	10	75.70	10	71.28	8
	PNM	70.66	10	77.49	10	82.72	10
20%							
	interpol.	24.36	10	65.51	10	70.01	10
	ADMM	69.86	10	67.09	3	69.62	10
	PGM	65.65	10	63.62	10	68.68	10
	FPGM	68.54	10	65.37	10	68.68	10
	PNM	64.76	10	66.22	10	70.61	10
30%							
	interpol.	−381	10	45.90	10	66.63	10
	ADMM	69.75	8	47.96	4	62.76	10
	PGM	65.78	10	46.09	10	63.30	10
	FPGM	66.33	10	46.21	10	63.66	10
	PNM	67.10	10	47.01	10	69.77	10
50%							
	interpol.	−58	10	22.71	10	27.69	10
	ADMM	56.85	10	33.66	10	19.81	10
	PGM	48.53	10	23.41	10	18.12	10
	FPGM	58.09	10	23.50	10	39.03	10
	PNM	53.31	10	28.94	10	50.92	10

Table 4.9: Missing outputs for CD player arm (left), thermic res wall (middle) and robot arm (right). The order is estimated with a threshold on singular values.

10%		fit	order	fit	order	fit	order
	interpol.	56.44	1	69.43	9	83.88	8
	ADMM	70.68	6	77.79	6	79.00	9
	PGM	69.54	10	72.55	2	67.83	8
	FPGM	69.45	7	70.77	8	67.83	8
	PNM	74.42	8	77.15	4	79.56	7
20%							
	interpol.	51.73	1	57.07	9	74.82	6
	ADMM	68.60	1	63.71	7	70.29	4
	PGM	66.04	1	62.29	2	71.36	10
	FPGM	68.54	2	62.31	5	71.36	10
	PNM	68.35	1	63.01	1	70.58	8
30%							
	interpol.	23.43	1	47.09	9	65.35	9
	ADMM	69.31	2	52.12	2	70.78	7
	PGM	67.79	8	50.41	3	69.81	10
	FPGM	68.38	1	49.97	5	70.09	10
	PNM	67.93	1	51.91	1	68.80	3
50%							
	interpol.	32.07	1	35.11	3	38.84	10
	ADMM	40.23	5	46.53	8	36.91	9
	PGM	39.05	2	34.58	4	33.06	3
	FPGM	37.85	3	34.59	4	33.11	3
	PNM	33.13	1	34.50	1	30.25	8

Table 4.10: Missing outputs for CD player arm (left), thermic res wall (middle) and robot arm (right). The order is estimated with the AIC principle.



(a) *ADMM*

(b) *FPGM*

(c) *PGM*

(d) *PNM*

(e) *interpolation*

Figure 4.11: Estimated outputs for CD player arm with 10% missing outputs. We notice that there are two outputs.

Appendix A

Computations

A.1 Gradient of $f^*(-\mathcal{A}_{adj}u)$

Let $F(u) := f^*(-\mathcal{A}_{adj}u)$ where f^* is the convex conjugate of a generic function f and $\mathcal{A}_{adj}$ is the adjoint of the mapping $\mathcal{A}$. We want to prove that

$$\nabla F(u) = -\mathcal{A}\nabla f^*(-\mathcal{A}_{adj}u)$$

We use the definition of convex conjugate:

$$\begin{aligned} F(y) &= f^*(-\mathcal{A}_{adj}y) = \sup_x\{\langle -\mathcal{A}_{adj}y, x\rangle - f(x)\} = \sup_x\{\langle -y, \mathcal{A}x\rangle - f(x)\} \\ &= -\inf_x\{\langle y, \mathcal{A}x\rangle + f(x)\} \end{aligned}$$

We remember the definition of subdifferential of F:

$$v \in \partial F(y) \quad\iff\quad F(z) \le F(y) + \langle v, z-y\rangle \quad \forall z$$

Now

$$F(z) = -\inf_x\{\langle z, \mathcal{A}x\rangle + f(x)\} = -\inf_x\{f(x) + \langle y, \mathcal{A}x\rangle + \langle z-y, \mathcal{A}x\rangle\}$$

We have

$$\begin{aligned} &\text{argmin}\{\langle \mathcal{A}x, y\rangle + f(x)\} = \text{argmax}\{-\langle \mathcal{A}x, y\rangle - f(x)\} \\ &\text{argmax}\{\langle x, -\mathcal{A}_{adj}y\rangle - f(x)\} = \nabla f^*(-\mathcal{A}_{adj}y) \end{aligned}$$

where the last equality is true for the properties of convex conjugate functions. Let $x^+ = \text{argmin}\{\langle \mathcal{A}x, y\rangle + f(x)\}$.

$$\underbrace{-\inf_x\{f(x) + \langle y, \mathcal{A}x\rangle + \langle z-y, \mathcal{A}x\rangle\}}_{=F(z)} \leq \underbrace{-(f(x^+) + \langle y, \mathcal{A}x^+\rangle + \langle z-y, \mathcal{A}x^+\rangle)}_{=F(y)-\langle z-y,\mathcal{A}x^+\rangle}$$

Then

$$F(z) \leq F(y) - \langle z-y, \mathcal{A}x^+\rangle = F(y) + \langle -\mathcal{A}x^+, z-y\rangle$$

Therefore,

$$-\mathcal{A}x^+ \in \partial F(y) \quad \text{If } F \text{ is differentiable} \quad -\mathcal{A}\nabla f^*(-\mathcal{A}_{adj}u) = \nabla F(y).$$

A.2 Adjoint of the mapping $\mathcal{A}$

Let $x = (h_0, h_1, ..., h_{r+N-2})$ and without loss of generality, we can suppose that $h_i \in \mathbb{R}\ \forall i = 1, ..., r+N-2$.

Let $\mathcal{A} : \mathbb{R}^{r+N-2} \to \mathbb{R}^{p\times q}$ the linear mapping in the nuclear norm optimization problem. It can be expressed as

$$\mathcal{A}(x) = L\mathcal{H}(x)R$$

where $L \in \mathbb{R}^{p\times(N-1)}$, $R \in \mathbb{R}^{(r-1)\times q}$ and $\mathcal{H}(x) \in \mathbb{R}^{(r-1)\times(N-1)}$ is a block Hankel matrix

$$\mathcal{H}(x) = \begin{bmatrix} h_0 & h_1 & \dots & h_{N-1} \\ h_1 & h_2 & \dots & h_N \\ \vdots & \vdots & \ddots & \vdots \\ h_{r-1} & h_r & \dots & h_{r+N-2} \end{bmatrix}$$

Then, the adjoint of the mapping $\mathcal{A}$ is $\mathcal{A}_{adj} : \mathbb{R}^{p\times q} \to \mathbb{R}^{r+N-2}$. For definition

$$\langle \mathcal{A}(x), y\rangle = \langle x, \mathcal{A}_{adj}(y)\rangle \implies \langle L\mathcal{H}(x)R, y\rangle = \langle x, \mathcal{H}_{adj}(L^TyR^T)\rangle$$
$$\forall x \in \mathbb{R}^{r+N-2}, y \in \mathbb{R}^{p\times q}$$

Now the adjoint $\mathcal{H}_{adj}$ of the Hankel mapping $\mathcal{H}$, maps an $(r-1)\times(N-1)$ matrix to a vector with $r+N-2$ components, by summing the entries in the matrix along the anti-diagonals. Hence we prove that

$$\langle \mathcal{H}_{adj}\mathcal{H}(x), x\rangle = \langle \mathcal{H}(x), \mathcal{H}(x)\rangle \quad \forall x \in \mathbb{R}^{r+N-2}$$

Here space is equipped with the trace inner product $\langle X, Y\rangle = Tr(X^TY)$. Without loss of generality we can suppose $r < N$. Therefore, from the definition:

$$\mathcal{H}_{adj}\mathcal{H}(x) = (h_0, 2h_1, ..., rh_{r-1}, (r-1)h_r, ..., h_{N+r-2})$$

$$\begin{aligned}
\langle \mathcal{H}_{adj}\mathcal{H}(x), x\rangle &= Tr(\mathcal{H}_{adj}\mathcal{H}(x)^T x) \\
&= \begin{bmatrix} h_0^2 & h_0h_1 & \dots & h_0h_{N+r-2} \\ 2h_0h_1 & 2h_1^2 & \dots & 2h_1h_{N+r-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_0h_{N+r-2} & h_1h_{N+r-2} & \dots & h_{r+N-2}^2 \end{bmatrix} \\
&= h_0^2 + 2h_1^2 + ... + rh_{r-1}^2 + ... + h_{N+r-2}^2
\end{aligned}$$

$$\begin{aligned}
&\langle \mathcal{H}(x), \mathcal{H}(x)\rangle = Tr(\mathcal{H}(x)^T\mathcal{H}(x)) \\
&= Tr\begin{bmatrix} h_0 & h_1 & \dots & h_{r-1} \\ h_1 & h_2 & \dots & h_r \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_N & \dots & h_{r+N-2} \end{bmatrix} \begin{bmatrix} h_0 & h_1 & \dots & h_{N-1} \\ h_1 & h_2 & \dots & h_N \\ \vdots & \vdots & \ddots & \vdots \\ h_{r-1} & h_r & \dots & h_{r+N-2} \end{bmatrix} \\
&= Tr\begin{bmatrix} h_0^2 + h_1^2 + ... + h_{r-1}^2 & & & \\ & h_1^2 + h_2^2 + ... + h_r^2 & & * \\ & * & \ddots & \\ & & & h_{N-1}^2 + h_N^2 + ... + h_{r+N-2}^2 \end{bmatrix} \\
&= h_0^2 + 2h_1^2 + ... + rh_{r-1}^2 + ... + h_{N+r-2}^2
\end{aligned}$$

In this way, we have proved that $\langle \mathcal{H}_{adj}\mathcal{H}(x), x\rangle = \langle \mathcal{H}(x), \mathcal{H}(x)\rangle$ $\forall x \in \mathbb{R}^{r+N-2}$. Therefore, the adjoint of mapping $\mathcal{A}$ is

$$\mathcal{A}_{adj}(y) = \mathcal{H}_{adj}(L^TyR^T) \quad \forall y \in \mathbb{R}^{p\times q}$$

A.3 DFP formula for updating H^k

The Davidon-Fletcher-Powell (DFP) formula is

$$B^{k+1} = B^k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{B^k y^k (y^k)^T B^k}{(y^k)^T B^k y^k}$$

where

- B^k is the approximation of the inverse of the Hessian,
- $s_k := x^{k+1} - x^k$,
- $y^k := \nabla f(x^{k+1}) - \nabla f(x^k)$

Let $U = [u_1, u_2]$ and $V = [v_1, v_2]$ with

$$u_1 = v_1 = \frac{s_k}{((y^k)^T s_k)^{1/2}}, \quad u_2 = -v_2 = \frac{B^k y^k}{((y^k)^T B^k y^k)^{1/2}}$$

We compute the matrix C of the Sherman-Morrison-Woodbury formula:

- $C_{11} = 1 + v_1^T (B^k)^{-1} u_1 = 1 + \frac{s_k^T H^k s_k}{(y^k)^T s_k} = \beta$
- $C_{22} = 1 + v_2^T (B^k)^{-1} u_2 = 1 - \frac{(y^k)^T B^k (B^k)^{-1} B^k y^k}{(y^k)^T B^k y^k} = 1 - 1 = 0$
- $C_{12} = 0 + v_1^T (B^k)^{-1} u_2 = \frac{s_k^T}{((y^k)^T s_k)^{1/2}} H^k \frac{B^k y^k}{((y^k)^T B^k y^k)^{1/2}} = \frac{((y^k)^T s_k)^{1/2}}{((y^k)^T B^k y^k)^{1/2}} = \alpha$
- $C_{21} = 0 - v_2^T (B^k)^{-1} u_1 = -C_{12} = -\alpha$

Then

$$C = \begin{bmatrix} \beta & \alpha \\ -\alpha & 0 \end{bmatrix}, \quad C^{-1} = \frac{1}{\alpha^2} \begin{bmatrix} 0 & -\alpha \\ \alpha & \beta \end{bmatrix}$$

Let $\tilde{U} = H^k U$ and $\tilde{V} = H^k V$, using Sherman-Morris-Woodbury formula

$$H^{k+1} = H^k - H^k U C^{-1} V^T H^k = H^k - \tilde{U} C^{-1} \tilde{V}^T$$

$$\tilde{U} C^{-1} \tilde{V}^T = \frac{1}{\alpha^2} [\tilde{u}_1, \tilde{u}_2] \begin{bmatrix} 0 & -\alpha \\ \alpha & \beta \end{bmatrix} \begin{bmatrix} \tilde{v}_1^T \\ \tilde{v}_2^T \end{bmatrix} =$$

$$= \frac{1}{\alpha} (H^k u_2 v_1^T H^k - H^k u_1 v_2^T H^k) + \frac{\beta}{\alpha} (H^k u_2 v_2^T H^k)$$

If we substitute the values of α, β, u_1, u_2, v_1 and v_2, we obtain

$$H^{k+1} = H^k - \frac{y^k s_k^T H^k + H^k s_k (y^k)^T}{(y^k)^T s_k} + \frac{y^k (y^k)^T}{(y^k)^T s_k}\left(1 + \frac{s_k^T H^k s_k}{(y^k)^T s_k}\right)$$

or equivalently

$$H^{k+1} = \left(I - \frac{y^k s_k^T}{(y^k)^T s_k}\right) H^k \left(I - \frac{s_k (y^k)^T}{(y^k)^T s_k}\right) + \frac{y^k (y^k)^T}{(y^k)^T s_k}$$

Bibliography

[1] Andersen M. S., Dahi J. Vandenberghe L. *CVXOPT Documentation, Release 1.2.3*, (2019) http://cvxopt.org/documentation

[2] Andersen M. S., Li J., Vandenberghe L. *Inexact proximal Newton methods for self-concordant functions*, Mathematical Methods of Operational Research 85(1), November 2016

[3] Antonello N., Stella L., Moonen M., Panagiotis P., van Waterschoot T. , *Proximal Gradient Algorithms: Applications in Signal Processing*, submitted to IEEE Signal Processing Magazine (2018)

[4] Beck A., Teboulle M., *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM J. Imaging Sciences, Vol. 2, No. 1, pp. 183-202

[5] Beck A., Teboulle M., *Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems*, IEEE Transactions on Image Processing, Vol. 18, No. 11 (2009)

[6] Bertolazzi E., *Quasi-Newton methods for minimization*, Lectures for PHD couse on Numerical optimization, DIMS- Università di Trento (2011), http://www.ing.unitn.it/ bertolaz/2-teaching/2011-2012/AA-2011-2012-OPTIM/lezioni/slides-mQN.pdf

[7] Boyd S., Parikn N., *Proximal Algorithms*, Foundations and Trends in Optimization, Vol. 1, No. 3 (2013) 123-231

[8] Boyd S., Chu E. Eckstein J., Parikn N., Peleato B., *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Foundations and Trends in Machine Learning, Vol. 3, No. 1 (2010) 1-122

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

[10] Byrd R. H., Nocedal J., Oztoprak F., *An inexact successive quadratic approximation method for convex L-1 regularized optimization*, Mathematical Programming 157(2) · September 2013

[11] De Moor B., De Gersem P., De Schutter B., Favoreel W., *DAISY: A database for the identification of systems*, Journal A 38 (3) (1997) 4-5

[12] Fazel M. Ting Kei Pong, Defeng Sun, Tseng P., *Hankel matrix rank minimization with applications to system identification and realization*, SIAM J. Matrix Anal. Appl., 34(3), 946–977

[13] Hansson A., Liu Z., Vandenberghe L., *Subspace System Identification via Weighted Nuclear Norm Optimization*, arXiv:1207.0023vl [cs.SY] (2012)

[14] Hansson A., Liu Z., Vandenberghe L., *Nuclear norm system identification with missing inputs and outputs*, Systems & Control Letters, Vol. 62, No. 8, Pages 605-612 (2013)

[15] Hansson A., Liu Z., Vandenberghe L., *SIMIO: Matlab package for system identification with missing inputs and outputs*, (2013) http://www.zhang-liu.com/software/file/simio.zip

[16] Lee J. D., Sun Y., Saunders M. A., *Proximal Newton-type methods for convex optimization*, Advances in Neural Information Processing System, pp. 836-844 (2012)

[17] Lee J. D., Sun Y., Saunders M. A., *Proximal Newton-type method for minimizing composite functions*, SIAM J.Optim. Vol. 24 No. 3, pp. 1420-1443

[18] Liu Z., Vandenberghe L., *Interior-point code for nuclear norm minimization (version 1.0)*, (2009) http://cvxopt.org/applications/nucnrm/

[19] Marcuzzi F., *Analisi dei Dati mediante Modelli Matematici*, http://www.math.unipd.it/ marcuzzi

[20] Nedich A., *Newton Method and Self-Concordance, Lecture 15*, (2008) Department of Industrial and Enterprise Systems Engineering, University of Illinois, http://www.ifp.illinois.edu/ angelia/L15_selfconcordant.pdf

[21] Tibshirani R., *Proximal Newton Method*, http://www.stat.cmu.edu/ ryantibs/convexopt-F15/lectures/17-prox-newton.pdf

[22] Vandenberghe L. *Convex optimization techniques in system identification*, , Electrical Engineering Department, UCLA, Los Angeles, CA 90095, http://www.seas.ucla.edu/ vandenbe/publications/sysid-244.pdf

[23] Vandenberghe L. *Conjugate functions*, lecture notes 7 for the course Optimization methods for large-scale systems, UCLA Electrical and Computer Engineering Department

[24] Verhaegen, Michel and Hansson, Anders, *N2SID: Nuclear norm subspace identification of innovation models*, Automatica, Vol. 72, No. C, pp. 57-63, October 2016

[25] Werner C. Rheinboldt, *Quasi-Newton Methods*, https://www-m2.ma.tum.de/foswiki/pub/M2/Allgemeines/SemWs09/quasi-newt.pdf