LUND UNIVERSITY
DEPARTMENT of MATHEMATICS

NUMK01

Model selection and optimization experiment of a field trial with agricultal data using Python

Author:
A. Lukas Bengtsson

Supervisor:
Claus Führer

March 31, 2018

Abstract

Potato cultivation is vast for many agricultures in Skåne. What many might not be aware of, is how much pesticides which are used in the cultivation process. In *"Late blight prediction and analysis"*[1] predictions of late-blight attacks was modelled on potato-corps using *Elastic Net* among other methods. This thesis is an algorithmic complement describing *Alternate Direction Method of Multipliers* (ADMM) and its use for efficient optimization of *Elastic Net*.

Initially a firm foundation is lied introducing the primal- respectively dual-problem and how the relation can be used to define optimization methods. The finale of those is ADMM. Its predecessors are also properly introduced in this thesis.

Then using the same data as in *"Late blight prediction and analysis"*[1], an experiment looking into parameter effects in both ADMM and *Elastic Net* is conducted. Some intuitions are confirmed, such as penalty effects and similar. Notably is the robustness of ADMM, despite this it can be more or less effective. Example wise it was found that some control parameters in ADMM and *Elastic Net* has a firm relations and should be properly chosen.

Contents

Chapter 1

Introduction

In order to preserve and optimize the harvest, farmers heavily use pesticides. One of the corps which needs the most spraying in Sweden is potatoes. Of those pests, late-blight is considered to be the worst. Late blight is a fungus like oomocyte which causes foliage decay. Due to its adaptiveness it is hard to find good persistent treatments. If farmers do not protect their harvest against late blight the expected profit will likely be heavily reduced. In addition to this, there is a climate trend which is beneficial to the late blight which is due to increasing humidity in Sweden [1].

The number of fungicidal sprays have been limited to reduce the environmental stress. Experiments suggest that spraying more than ten days before the first sign of late blight still obtain an optimal effect, but if the spraying is more delayed relative losses are very probable. After the first application the farmers continuously apply fungicides approximately once a week until defoliation[1]. One of the large issues is to predict the appearance date which is needed in order to spray in time. The paper *"Late blight prediction and analysis"* is an experiment trying to find more accurate models which can be used to forecast the first sign of late blight [1].

[1] Before harvest the plant foliage is chemically removed which is known as defoliation.

The data which was used in the thesis *"Late blight prediction and analysis"* included cultivation variables such as fertilization need on the fields, kind of soil, irrigation, amount of fertilization, pH value, location of experiment, year, treatments, late blight infection levels planting-, defoliation- and harvest dates. These data was collected in three locations, Kristianstad-Mosslunda, Borgeby and Lilla Eldsberga during 1983-2012.

due to adjusted law regulations, the adaptiveness of the oomycyte and several other factors, the treatments used in the experiments are different through the years and only denoted as "best", "worst", "control group to best" and "non-treated". The weather data was modeled to corresponding station in lack of real measurements. It contained hourly samples of wind speed, temperature, rain fall, relative humidity and relative humidity above 90%. Several data transformations were considered manipulating the covariates in order to improve the result. Lastly missing data was restored using imputation [1].

All of above was considered to be possible parameters, which resulted up in approximately 70 different variables. Some measurements were gathered yearly, some daily and a few hourly. Due to change in late blight dynamics only data between 1994-2012 was used resulting in 54 measures of late blight infections. These measurements were gathered in a $p \times n$ matrix A where the columns consists of kind o measurements and rows the value at given time, location and treatment. To obtain a robust consistent model, parameter estimation in the experiment was made using elastic net and manual parameter selection. The models considered was linear regression and cox-regression validated by cross-validation [1].

In *"Late Blight Prediction and Analysis"* the methodology is discussed where as many of the numerical aspects were left out [1]. This paper is intended as a complement to the paper to further investigate one of the possible numerical approaches, namely problem solving using *Elastic Net* by *Alternating Direction Method of Multipliers* (ADMM) optimization. The main emphasis in this thesis lies within experimentation of parameter choices in ADMM respectively *Elastic Net* by investigating how these affect convergence, accuracy and the final model.

To get a deeper understanding of ADMM optimization also its precursors, the *Dual Ascent* method and *Method of Multipliers*, will be studied in this thesis. The thesiMethods will be restricted to one of the issues which was looked into in *"Late Blight Prediction and Analysis"*, namely profit modeling. All experiments will be carried out using Python.

1.1 Optimization methods

Alternating Direction Method of Multipliers, also known as ADMM, can be traced back to the classic paper *On the numerical solution of heat conduction problems in two and three space variables* which was published 1956 [2]. Since then multiple papers have been published in the subject and still up-to-date. During early stages several papers with similar methods were developed and eventually equivalence was noted between them. One such case is the *Douglas-Rachford* algorithm known from numerical analysis which is equivalent to ADMM [3].

The efficiency of ADMM depends heavily on the problem setting. Example wise under strong convexity of one objective term it can be shown that it has linear convergence [4, 5], whereas in the most general setting the convergence rate is $1/\epsilon^2$ [6]. ADMM is not always the best algorithmic choice as there may be more efficient methods under some conditions. But ADMM preforms rather well in spite of the problem setting and is very flexible. Currently the methodology is used in many machine-learning and big data applications [3].

1.1.1 The Primal and Dual problems

The primal problem

Definition 1. *For an objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, with linear constraints $Ax = b$, where A is an $m \times n$ matrix of rang m with $m < n$, and b a column vector with m elements. Then we say the primal problem is:*

$$\begin{aligned}&\textit{minimize } f(x) \textit{ with respect to } x\\ &\textit{subject to } Ax = b.\end{aligned} \tag{1.1}$$

Note that the extreme-values of f are not necessarily equal to those under the constraints.

Feasibility

Any vector $\hat{x}$ which satisfies the given constraints is said to be a feasible point. The set of all vectors satisfying the constraints is know as the feasibility region. For any pair of feasible points $\hat{x} \neq \bar{x}$ the following must be satisfied:

$$A\hat{x} = A\bar{x} = b$$

$$\iff A(\hat{x} - \bar{x}) = 0^{(m \times 1)}.$$

where $0^{(m \times 1)}$ is a zero vector with dimension $(m \times 1)$. We call $p = \hat{x} - \bar{x}$ a feasible direction i.e.

$$A(\hat{x} + \alpha p) = A\hat{x} + \alpha A(\hat{x} - \bar{x}) = A\hat{x} = b$$

is feasible for all scalars $\alpha \in \mathbb{R}$ [7].

Optimality conditions of the primal problem

There always exists an $n \times m$ matrix Z satisfying $AZ = 0^{(m \times m)}$. Then each column in Z must be a feasible direction as $AZ_{:,i} = 0^{(m \times 1)}$ for all $i = 1, 2, \ldots, m$. Similarly for all columns that span the space of feasible directions.

Consider an arbitrary linear combination of Z, $p = Zp_z$. Further denote the gradient of f as ∇f and its Hessian as H, moreover let $\epsilon \in \mathbb{R}$ and $0 \leq \theta \leq 1$. Then by Taylor expanding f along p we obtain:

$$f(x^* + \epsilon Z p_z) = f(x^*) + \epsilon p_z^T Z^T \nabla f(x^*) + \frac{1}{2}\epsilon^2 p_z^T Z^T H(x^* + \epsilon p \theta) Z p_z.$$

Note that if:

$$\epsilon p^T Z^T \nabla f(x^*) < 0 \tag{1.2}$$

then if $|\epsilon|$ is small enough

$$\epsilon p_z^T Z^T \nabla f(x^*) + \frac{1}{2}\epsilon^2 p_z^T Z^T H(x^* + \epsilon p \theta) Z p_z < 0$$

$$\implies f(x^* + \epsilon p) < f(x^*).$$

I.e. in a neighbourhood of x^* exists a point with lower function value, hence if (1.2) is true x^* can not be a minimum. This implies that if x^* solves the primal problem (1.1) then $p_z Z^T \nabla f(x^*) = 0$. As the same holds for all linear combinations p, we further obtain if x^* is a solution then [7]:

$$Z^T \nabla f(x^*) = 0^{(n \times 1)}. \tag{1.3}$$

As A has full rank and $m < n$, $\mathbb{R}^n = \ker(A) \oplus \operatorname{im}(A^T)$. Consequently there exists λ, f_z such that

$$\mathbb{R}^n \ni \nabla f(x^*) = A^T y + Z f_z. \tag{1.4}$$

Multiplying Equation (1.4) with Z^T from the left we obtain:

$$Z^T \nabla f(x^*) = Z^T A^T y + Z^T Z f_z.$$

Equation (1.3) implies that if x^* is a solution to the primal problem then $Z^T \nabla f(x^*) = 0^{(n \times 1)}$. Further the properties of a basis implies that $Z^T Z$ is non-singular. Hence f_z in Equation (1.4) must be a zero vector. Then it follows from Equation (1.4) that the gradient $\nabla f(\hat{x})$ is a linear combination of the rows of A:

$$\nabla f(\hat{x}) = \sum_{i=1}^{n} y A_{i,:} = y^T A$$

for some $m \times 1$ vector y. The vector y is uniquely determined only if the rows of A are linearly independent [7].

If we consider the Taylor expansion along the feasible direction p, assuming x^* solves the primal problem (1.1) we get:

$$f(x^* + \epsilon p) = f(x^*) + \frac{1}{2}\epsilon^2 p_z^T Z^T H(x^* + \epsilon p \theta) Z p_z.$$

If the projected Hessian is negative definite $Z^T H(x^* + \epsilon p\theta) Z < 0$, then $f(x^* + \epsilon p) < f(x^*)$. This contradicts the fact that it is a solution to the primal problem. Hence in order for x^* to be a solution, the projected Hessian must be positive semi-definite [7].

This can be used as a tool to solve a large problem in a smaller setting. In summary the optimality conditions for a feasible $\hat{x}$ are[7, 3]:

c.1 $A\hat{x} - b = 0$

c.2 $\nabla f(\hat{x}) - y^T A = 0$

c.3 projected Hessian must be positive semi-definite.

The Lagrangian

For a given primal problem (1.1) the Lagrangian is defined as:

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

where y is the $(1 \times n)$ vector of Lagrangian multipliers and f the objective function which are constrained by $Ax = b$. Then setting the partial derivative with respect to y, to zero will ensure (c.1),

$$\frac{\partial L(x, y)}{\partial y} = Ax - b = 0$$

and similar the derivative with respect to x:

$$\frac{\partial L(x, y)}{\partial x} = \nabla f(x) + y^T A = 0$$

becomes the second optimality condition (c.2). We emphasize, that for all feasible $\hat{x}$ (c.1) implies:

$$L(\hat{x}, y) = f(\hat{x}) + y^T (A\hat{x} - b) = f(\hat{x}).$$

and similarly

$$\min_{\hat{x} \in p} \ f(\hat{x}) = \min_{\hat{x} \in p} \ L(\hat{x}, y).$$

The dual problem

Definition 2. *Denote the Lagrangian as $L(\omega, y)$, $\omega \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Then the dual problem is defined as [8]:*

$$\begin{aligned} &\textit{maximize } L(\omega, y) \textit{ with respect to } y \textit{ and } \omega \\ &\textit{subject to } \frac{\partial L(\omega, y)}{\partial \omega} = 0. \end{aligned} \tag{1.5}$$

Further it can be shown that the dual objective function must have a negative definite Hessian [9]. The dual constraints imply minimization with respect to ω, hence dual problem can be equivalently expressed as $\max_y \min_\omega L(\omega, y)$. Then the dual objective function is identified as $g(y) = \min_\omega L(\omega, y)$.

Example

Let the objective function f represent a cost of some goods and the constraints rules regarding the goods. Rule violations are penalized with an additional cost y. The total cost would be given by the Lagrangian.

The primal problem would be to minimize the cost without violations. The Dual problem is to maximize the cost by maximizing the violation penalty along feasible directions.

Dual gap

Let x^* be a solution to the primal problem and y^* to the corresponding dual problem. Then we call the difference $f(x^*) - L(x^*, y^*)$ the dual gap. If the dual gap equals zero, it is said strong duality holds. Assuming convexity of f and strong duality, the solution to the primal problem, x^*, can be recovered by [3]:

$$x^* = \min_x \quad L(x, y^*) \tag{1.6}$$

This is the engine which will be used in the optimization algorithms considered in this thesis.

1.1.2 The Dual Ascent & Dual Decomposition methods

Given a primal problem (1.1) with the linear constraints $Ax = c$. Under the assumptions of strict convexity i.e.

$$f(tx_1 + (1 - tx_2)) < tf(x_1) + f((1-t)x_2) \quad \forall x_1 \neq x_2,\ \forall t \in (0,1)$$

and differentiability of the objective function f, the *Dual Ascent* method consists of the following iteration [10]:

$$x^{k+1} = \text{argmin}_x \quad L(x, y^k)$$
$$y^{k+1} = y^k + \rho(Ax^{k+1} - c)$$

for given step size $\rho > 0$. By maximizing its dual function[2] $g(y)$ it converges to the solution of the primal problem [3]. The iterates should proceed until primal- and dual feasibility are obtained for some set tolerances.

If the objective function f is separable, i.e. it satisfies:

$$f(x) = \sum_{i=0}^{p} f_i(x_i)$$

then it is possible to solve partial sub-problems in parallel, this is know as the *Dual Decomposition* [3, 11].

[2] Ascending the dual of which the name Dual Ascent.

1.1.3 The Method of Multipliers and augmented Lagrangian

The proximal operator

A function $h : \mathbb{R}^n \to \mathbb{R}$ is called a closed if

$$\{x \in \text{ dom} f | f(x) \leq \mu\} \ \forall \mu \in \mathbb{R}$$

is a closed set. If $h(x)$ is convex, $h(x) > -\infty \ \forall x$ and h is not identically equal ∞, the function is proper convex.

Assume $h : \mathbb{R}^n \to \mathbb{R}$ is a proper closed convex function then the proximal is defined as:

$$\text{prox}_h(v) = \text{argmin}_x \left(h(x) + \frac{1}{2\rho} ||x - v||_2^2 \right), \qquad \rho > 0$$

This can be interpreted as a trade off between minimizing h and being close to v for some weight ρ. It can be shown that [12]:

$$x^* = \text{argmin } h(x) \quad \iff \quad \text{prox}_h(x^*) = x^*,$$

Hence it is possible to find the optimum of a function h using the proximity operator. One of the most primitive methods to do so is the proximal point algorithm which is defined as:

$$x^{k+1} = \text{prox}_h(x^k). \tag{1.7}$$

The augmented Lagrangian

Applying the proximal point algorithm (1.7) to the dual function we obtain:

$$\text{prox}_g(y^k) = \text{max}_y \text{argmin}_x \left(L(x, y) - \frac{1}{2\rho} ||y - y^k||_2^2 \right).$$

Maximizing with respect to the dual variable, y, we find:

$$Ax - b - \frac{1}{\rho}(y - y^k) = 0$$
$$\iff y^k = y - \rho(Ax - b).$$

Then substituting into the original problem:

$$\text{max}_y \text{argmin}_x \left(L(x, y) - \frac{1}{2\rho} ||y - y^k||_2^2 \right) = \text{max}_y \text{min}_x \left(f(x) + y^T(Ax - b) + \frac{\rho}{2} ||Ax - b||_2^2 \right).$$

The minimized function is known as the augmented Lagrangian, $L_\rho(x, y)$, i.e.:

$$L_\rho(x, y) = L(x, y) - \frac{\rho}{2} ||Ax - b||_2^2$$

The Method of Multipliers

In order to increase the robustness and avoid the strict convexity assumption of f, the *Method of Multipliers* was developed [3]. It extends the *Dual Ascent* method by considering the augmented Lagrangian:

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

where $\rho > 0$ can be viewed as a penalty parameter. Note that if $\rho = 0$ the augmented Lagrangian would coincide with the Lagrangian. The *Method of Multipliers* considers the reformulated primal problem:

Minimize $f(x) + \rho/2||Ax - c||_2^2$ with respect to x

Subject to $Ax = c$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{p\times n}$ and $f : \mathbb{R}^n \to \mathbb{R}$. Note that for feasible $\hat{x}$, $f(\hat{x}) + \rho/2||A\hat{x} - c||_2^2 = f(\hat{x})$. Following the same structure as in *Dual Ascent*, assuming the duality gap being zero, the optimal solution x^* may be obtained as:

$$x^* = \text{argmin}_x L_\rho(x, y^*)$$

where y^* is the optimal dual solution. Then the *Dual Ascent for augmented Lagrangian*, also known as the *Method of multipliers*, can be formulated as:

$x^{k+1} = \text{argmin}_x L_\rho(x, y^k)$

$y^{k+1} = y^k + \rho(Ax^k - c)$

where the scalar ρ can be interpreted as step size. By using the penalty parameter as step size, convergence to a feasible region is ensured. It's possible to use non-constant ρ to improve the convergence, however in this thesis it is assumed to be constant. The *Method of Multipliers* has better convergence properties and need less assumptions compared to the *Dual Ascent*. In particular it can be applied in cases where f is not strictly convex. A huge con is as the augmented Lagrangian is not generally separable, the method can not be decomposed [3].

1.1.4 Alternating Direction Method of Multipliers

ADMM is known to be a mixture between *Dual Ascent* and *Method of Multipliers*, taking the best from both algorithms. Decomposability from *Dual Ascent* and the convergence properties of *Method of Multipliers.* Assuming f and g are convex with the constraint $Ax + Bz = c$, the following primal problem is considered:

$$\text{Minimize } f(x) + g(z)$$

$$\text{Subjective to } Ax + Bz = c$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$. Note that introducing the secondary variable z the method becomes more flexible. Similar to the method of multipliers, it uses the augmented Lagrangian which extends to:

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

in the ADMM case [3].

By alternating updates between x and z followed by a dual update the ADMM algorithm is [3]:

$$x^{k+1} = \min_x \quad L_\rho(x, y^k, z^k)$$

$$z^{k+1} = \min_z \quad L_\rho(x^k, y^k, z)$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c).$$

Stopping criteria

Based on the primal and dual feasibility regions, stopping criterion can be derived. In an ADMM setting those are:

primal feasibility:

$$Ax^* + Bz^* - c = 0$$

dual feasibility:

$$\frac{\partial f(x^*)}{\partial x^*} + A^T y^* \in 0, \text{and} \qquad \frac{\partial g(x^*)}{\partial x^*} + B^T y^* \in 0.$$

Based on these the dual (ϵ_{dual}) and primal (ϵ_{primal}) residuals can be derived, see [3, Appendix A] for a complete derivation:

$$r_{primal} = Ax^k + Bz^k - c.$$

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

If these are small enough convergence has been obtained. One way to decide whether they are good enough is to use the absolute- and relative tolerances, $\epsilon_{Absolute}$ respectively $\epsilon_{Relative}$:

$$\epsilon_{primal} = \epsilon_{Absolute}\sqrt{p} + \epsilon_{Relative}\text{max}\left(||Ax^k||_2^2, ||Bz^k||_2^2, ||c||_2^2\right)$$

$$\epsilon_{dual} = \epsilon_{Absolute}\sqrt{n} + \epsilon_{Relative}||A^T y^k||_2^2.$$

Then the stopping criterion is:

$$\text{Stop if} : \begin{cases} ||r^k_{primal}|| \leq \epsilon_{primal} \\ ||r^k_{dual}|| \leq \epsilon_{dual} \end{cases} . \tag{1.8}$$

See *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* [3, Appendix A] for details.

Convergence

Assuming $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ respectively $g : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ are closed proper and convex. Further assuming the unaugmented Lagrangian has a saddle point, it can be shown that:

- The residual, $r = Ax + Bz - c$, converge to zero. I.e. the problem tends to feasible solutions. $$r^k \xrightarrow[k\to\infty]{} 0$$
- The objective function converge to the optimal value of the primal problem. $$f(x^k) + g(z^k) \xrightarrow[k\to\infty]{} p^*$$
- The dual variable converges to its optimal value. $$y^k \xrightarrow[k\to\infty]{} y^*$$

Neither x or z necessarily converge to their optimal values under the given assumptions, however they do get arbitrarily close [3].

In practice ADMM preform rather well under modest accuracy conditions, but tends to be slow if higher precision is wanted. Its possible to combine ADMM with other algorithms[3] in order to improve convergence ratios etc. In general ADMM is suitable for large-scale problems where modest accuracy is enough [3].

[3] This will not be investigated in this thesis.

1.1.5 Overview

To summarize the methods we consider the ADMM algorithm as a basis. Let the penalty parameter from *Method of Multipliers* and *ADMM* be ρ_1 and call the step size parameter ρ_0. Then consider the following set up:

$$
\begin{aligned}
L_\rho(x, y, z) &= f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho_1}{2}||Ax + Bz - c||_2^2 \\
x^{k+1} &= \min\nolimits_x \quad L_\rho(x, z^k, y^k) \\
z^{k+1} &= \min\nolimits_z \quad L_\rho(x^k, z, y^k) \\
y^{k+1} &= y^k + \rho_0(Ax^{k+1} + Bz^{k+1} - c)
\end{aligned}
$$

Then we obtain by respective choice:

- ADMM — if $\rho_0 = \rho_1$
- Method of Multipliers — if $\rho_0 = \rho_1$, $g(z) = 0$ and $Bz = 0$, i.e. joint minimization of all variables.
- Dual Ascent — if $\rho_1 = 0$, $g(z) = 0$ and $Bz = 0$
- Dual Decomposition — if $\rho_1 = 0$, $g(z) = 0$, $Bz = 0$ and the objective function is decomposed.

Note that above is only from an algorithmic point of view. Example wise is the *Dual Ascent* more restricted by assumptions such as strict convexity.

1.2 Modelling Data

When creating models several issues appear, which kind of model is the most suitable, which parameters should be used, is it necessary to consider transformations etc. Given observations, $a = [a_1, a_2, \ldots, a_n]$, we want a model of the following form:

$$f(a) = \sum_{i=1}^{n} x_i a_i + \epsilon_i$$

where $f(a)$ is the response as a function of given observations a with parameters $x = [x_1, x_2, \ldots, x_n]$ and ϵ_i noise. Assume we are given n observations of a, denote these as the rows in the matrix A, and respective response $c = [c_1, c_2, \ldots, c_m]$. A common approach to such problem would be to find the x which minimize the mean residual $x^{(LS)} = \text{argmin}_x ||Ax - c||_2^2$ and then use the estimate of x, $x^{(LS)}$ to estimate the response:

$$f(\hat{a}) = \sum_{i=1}^{p} x_i^{(LS)} a_i.$$

The parameter estimation is know as the least squares (LS) estimate. Such approach does not account for some measurements may actually not be related to the response. This is known as over-fitting, i.e. some parameters would try to estimate noise after all actual information is depleted. If such occur the model would likely preform worse with data which was not used to estimated the model. Consequently there might exist better models which exclude some of the available covariates. In order to cope with such issues several methods can be used. Commonly used are the Akaike's and Bayesian information criteria which evaluate the maximum likelihood relative the information gained by each parameter. These methods does not preform very well if the parameter space is large relative the measurements [13].

1.2.1 Lasso and Elastic Net

The Least Absolute Selection and Shrinkage Operator, Lasso in short, have a different way to approach over-fitting issues. It aims to find the x which minimizes:

$$x^{(Lasso)} = \text{argmin}_x \frac{1}{2}||Ax - b||_2^2 + \lambda||x||_1 \tag{1.9}$$

where $||x||_1 = \sum_{i=1}^{p} |x_i|$, $x = [x_1, x_2, \ldots, x_p]$, is the l_1-norm and λ is a given penalty. Adding the l_1-norm ensures the problem becomes convex and that the estimate can include zero-valued parameters (those can be interpreted as removed covariates). The amount of excluded parameters is adjusted using the penalty λ [13].

If the data is heavily correlated the lasso rarely perform as well as the generalization *Elastic Net*:

$$x^{(Enet)} = \text{argmin}_x \frac{1}{2}||Ax - b||_2^2 + \lambda \left(\alpha||x||_1 + (1 - \alpha)\frac{1}{2}||x||_2^2 \right) \tag{1.10}$$

which allows an arbitrary mix $\alpha \in [0, 1]$, of the l_1 and l_2 penalties. If $\alpha \leq 1$ and $\lambda > 0$ the problem is strictly convex [13]. When $\alpha = 0$, i.e. only penalization using the l_2-norm, it is known as Ridge regression [14]. Note that no parameter will be removed when excluding the l_1-penalty .

For simplicity let $\lambda_1 = \alpha\lambda$ and $\lambda_2 = (1 - \alpha)\lambda$. The *Elastic Net* problem can equivalently be formulated as [3]:

$$\begin{cases} \text{Minimize } \frac{1}{2}||Ax - b||_2^2 + \lambda_1||z||_1 + \frac{1}{2}\lambda_2||x||_2^2 \\ \text{subject to x=z} \end{cases} \tag{1.11}$$

As this formulation of *Elastic Net* fits the ADMM framework (1.1.4) very well, it will be used through out the thesis. Identifying parts from the ADMM framework we find

$$\mathbf{A} = -B = \mathbb{I},\ c = 0$$

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

$$g(z) = \lambda_1||z||_1 + \frac{1}{2}\lambda_2||z||_2^2.$$

Note that $\mathbf{A}$ denotes the constraints in ADMM and does not equal A.

1.3 Sub-differentiability

As some functions which are considered in this thesis are not differentiable, we shortly introduce the concept of sub-differentials. For a function $g : \mathbb{R}^n \to \mathbb{R}$ we say c is a sub-gradient if it satisfies:

$$g(z) \geq g(z_0) + c(z - z_0), \quad \forall z. \tag{1.12}$$

The sub-differential in z_0, $\partial g(z_0)$, is the set of all sub-gradients c. If g is differentiable at z_0, the sub-differential $\partial g(z_0)$ equal the gradient $\nabla g(z_0)$. Some useful algebraic properties the sub-differential satisfies are:

$$\partial g(\alpha z) = \alpha \partial g(z) \tag{1.13}$$

$$\partial(g_1 + g_2) = \partial g_1 + \partial g_2. \tag{1.14}$$

Additionally if g is convex then it satisfies [15]

$$z^* = \operatorname{argmin}_z \; g(z) \iff 0 \in \partial g(z^*).$$

As an example consider $g(z) = |z|$, then the sub-differential of around z_0 is found as:

$$\begin{aligned}
\text{if } z_0 = 0\text{:} \quad & |z| - |0| \geq c(z - 0) && \forall z \\
& \iff 1 \geq \frac{cz}{|z|} && \forall z \\
& \implies c \in [-1, 1] \\
\\
\text{else if } z_0 < 0\text{:} \quad & |z| - |z_0| \geq c(z - z_0) && \forall z \\
& \iff |z| \geq cz + z_0(c + 1) && \forall z \\
& \implies c = -1 \\
\\
\text{else if } z_0 > 0\text{:} \quad & |z| - |z_0| \geq c(z - z_0) && \forall z \\
& \iff |z| \geq cz + z_0(c - 1) && \forall z \\
& \implies c = 1
\end{aligned}$$

obtaining the sub-differential of f:

$$\partial|z| = \begin{cases} \{-1\} & \text{if } z < 0 \\ [-1, 1] & \text{if } z = 0 \\ \{1\} & \text{if } z > 0. \end{cases} \tag{1.15}$$

Chapter 2

Analysis of parameter effect

2.1 Derivation of Elastic Net optimized by ADMM and its precursors

To use any of the dual algorithms either the Lagrangian or augmented Lagrangian is minimized with respect to x respectively z. Initially we consider *Method of Multipliers* where joint minimization is needed. Then we deduce it to *Dual Ascent*. For minimization in two variables both partial derivatives must jointly equal zero. First we consider the derivative with respect to x. Using the notation $\lambda_1 = \lambda\alpha$ and $\lambda_2 = \lambda(1-\alpha)$, the x-update is found as[1]:

$$\begin{aligned} &\text{argmin}_x \quad L_\rho(x, y, z) = \\ = \text{argmin}_x \quad &\frac{1}{2}||Ax - c||_2^2 + \lambda_1||z||_1 + \frac{\lambda_2}{2}||z||_2^2 + y^T(x - z) + \frac{\rho}{2}||x - z||_2^2 \end{aligned}$$

[1] Note that update indices are disregarded for clearer derivations. ”You might not see the forest because all of the trees”.

$$= \text{argmin}_x \quad \frac{1}{2}(x^TA^TAx - 2x^TA^Tc + c^Tc) + \frac{\lambda_2}{2}z^Tz + y^Tx - y^Tz + \frac{\rho}{2}(x^Tx - 2x^Tz + z^Tz)$$

$$= \text{argmin}_x \quad \frac{1}{2}(x^TA^TAx - 2x^TA^Tc) + y^Tx + \frac{\rho}{2}(x^Tx - 2x^Tz)$$

$$\iff A^TAx - A^Tc + y + \rho(x - z) = 0$$

$$\iff x = (A^TA + \rho\mathbf{I})^{-1}(A^Tc + \rho z - y)$$

Next minimizing with respect to z:

$$\text{argmin}_z \quad L_\rho(x, y, z) =$$

$$\text{argmin}_z \quad \frac{1}{2}||Ax - c||_2^2 + \lambda_1||z||_1 + \frac{\lambda_2}{2}||z||_2^2 + y^T(x - z) + \frac{\rho}{2}||x - z||_2^2$$

$$= \text{argmin}_z \quad \lambda_1||z||_1 + \frac{\lambda_2}{2}z^Tz - y^Tz + \frac{\rho}{2}(z^Tz - 2z^Tx)$$

Note that $||z||_1$ is not differentiable, therefore we consider the sub-differential. Recall the additive and scaling properties (1.14, 1.13) of the sub-differential. Consequently of these we get the decomposition $\partial\lambda||z||_1 = \partial\lambda\sum_{i=1}^p |z_i| = \lambda\sum_{i=1}^p \partial|z_i|$. Moreover if the sub-differential is differentiable it equals the gradient i.e.

$$\frac{\partial L_\rho(x, y, z)}{\partial z} = \frac{\partial\lambda_1||z||}{\partial z} + \nabla_z\left(\frac{\lambda_2}{2}z^Tz - y^Tz + \frac{\rho}{2}(z^Tz - 2z^Tx)\right)$$

$$= \frac{\partial\lambda_1||z||}{\partial z} + \lambda_2 z - y + \rho(z - x)$$

Recall Equation (1.15) in Section (1.3) and consider each z_i individually:

$$\frac{\partial L_\rho(x_i, y_i, z_i)}{\partial z_i} = \begin{cases} -\lambda_1 + \lambda_2 z_i - y_i - \rho x_i + \rho z_i & if \quad z_i < 0 \\ [-\lambda_1 - (y_i + \rho x_i), \lambda_1 - (y_i + \rho x_i)] & if \quad z_i = 0 \\ \lambda_1 + \lambda_2 z_i - y_i - \rho x_i + \rho z_i & if \quad z_i > 0 \end{cases}$$

The minimum of $f(z_i) = |z_i|$ is obtained if 0 ∂f, i.e.:

$z_i = 0,$

$$[-\lambda_1 - y_i - \rho x_i, \lambda_1 - y_i - \rho x_i] \in 0$$
$$\implies z_i = 0 \qquad \text{if } y_i + \rho x_i \in [-\lambda_1, \lambda_1]$$

$z_i < 0,$

$$-\lambda_1 + \lambda_2 z_i - y_i - \rho x_i + \rho z_i = 0$$
$$\implies z_i^* = \frac{\lambda_1 + y_i + \rho x_i}{\lambda_2 + \rho}, \qquad y_i + \rho x_i < -\lambda_1$$

$z_i > 0,$

$$\lambda_1 + \lambda_2 z_i - y_i + \rho x_i - \rho z_i = 0$$
$$\implies z_i^* = \frac{-\lambda_1 + y_i + \rho x_i}{\lambda_2 + \rho}, \qquad y_i + \rho x_i > \lambda_1$$

A more compact form of the solution (z_i^*) is:

$$z_i^* = \left(\frac{y_i + \rho x_i - \lambda_1}{\lambda_2 + \rho}\right)_+ - \left(\frac{-y_i - \rho x_i - \lambda_1}{\lambda_2 + \rho}\right)_+ \tag{2.1}$$

where $h(a)_+ = max(0, a)$ i.e. the positive part operator [3]. Due to the dependence in the joint x and z minimization, i.e. one is needed to obtain the other, the problem becomes difficult. It is possible to solve using combinatorial methods, but it would add further calculations in each iteration rapidly adding up. A few different approaches to avoid this issue has been tested all ending in failure.

2.1.1 The Dual Decomposition method

Consider the *Dual Ascent* ($\rho = 0$) method, then the x and z updates will be independent:

$$\begin{aligned} &\text{argmin}_x = (A^T A)^{-1}(A^T c - y) \\ \iff \quad &\partial(\lambda||z||_1) + \lambda_2 z - y \in 0 \end{aligned}$$

which is possible to solve. As the l_1-norm is separable and the x, z updates are independent we use the *Dual decomposition.* Then we obtain the z_i update:

$$\iff \partial(\lambda_1|z_i|_1) + \lambda_2 z_i + y_i \in 0 \qquad \forall i = 1, \ldots, p.$$

As this is a special case of Equation (2.1), we obtain the solution:

$$\iff z_i = \left(\frac{y_i - \lambda_1}{\lambda_2}\right)_+ - \left(\frac{-y_i - \lambda_1}{\lambda_2}\right)_+ \qquad \forall i = 1, \ldots, p.$$

2.1.2 The Alternating Method of Multipliers

If we allow alternating updates, i.e. use the ADMM algorithm, it is possible to proceed where the *Method of Multipliers* got stuck. Obtaining the z_i-update:

$$z_i = \left(\frac{\rho x_i + y_i - \lambda_1}{\lambda_2 + \rho}\right)_+ - \left(\frac{-\rho x_i - y_i - \lambda_1}{\lambda_2 + \rho}\right)_+ \qquad \forall i = 1, \ldots, p.$$

and, derived as in *Method of Multipliers*, the x-update:

$$x^{k+1} = (A^T A + \rho\mathbf{I})^{-1}(A^T c + \rho z - y).$$

ADMM is often given in the scaled form where $u = y/\rho$, resulting in the algorithm:

$$\begin{aligned} x^{k+1} &= (A^T A + \rho\mathbf{I})^{-1}(A^T c + \rho(z^k - u^k)) \\ z_i^{k+1} &= \left(\frac{\rho(x_i^{k+1} + u_i^k) - \lambda_1}{\lambda_2 + \rho}\right)_+ + \left(\frac{-(\rho x_i^{k+1} + u_i^k) - \lambda_1}{\lambda_2 + \rho}\right)_+ \\ u^{k+1} &= u^k + (x^k - z^k) \end{aligned}$$

ADMM optimized Elastic Net stopping criterion

Recall the general form of the stopping criterion (1.8). Using the *Elastic Net* as objective function we obtain as before [3]:

$$\mathbf{A} = -B = \mathbb{I}$$

$$c = 0.$$

Assume we want to define a stopping criterion based on the relative residual $\epsilon_{Relative}$ and the constant error $\epsilon_{Absolute}$. As suggested in section (1.1.4) the primal respectively dual residuals can be obtained as:

$$r_{primal} = \epsilon_{Absolute}\sqrt{p} + \epsilon_{Relative}\ max(||x||_2^2, ||-z||_2^2)$$

$$r_{dual} = \epsilon_{Absolute}\sqrt{n} + \epsilon_{Relative}||-\rho y)||_2^2.$$

and their feasibility criteria:

$$\epsilon_{primal} = ||x^k - z^k||_2^2$$

$$\epsilon_{dual} = ||-\rho(z^k - z^{k-1})||_2^2$$

Using above we obtain the stopping criterion as follows:

Stop if

$$(r_{primal} \leq \epsilon_{primal} \quad \cap \quad r_{dual} \leq \epsilon_{dual}) == \text{True}$$

Else proceed iterations.

2.1.3 Elastic Net Algorithm using ADMM optimization

Input and output

The combined ADMM and *Dual Decomposition* algorithm will have the inputs:

Method	:	Dual Decomposition or ADMM
A	:	$p \times n$ matrix of covariate observations
b	:	response vector with length n
λ	:	penalty ($\lambda \geq 0$)
α	:	penalty weight ($\alpha \in [0, 1]$)
ρ	:	step size and augmentation penalty ($\rho \geq 0$)
ϵ_{Abs}	:	absolute tolerance
ϵ_{Rel}	:	relative tolerance
MaxIter	:	maximum number of iterations allowed

The algorithm estimates the *Elastic Net* objective function,

$$f(x) = ||Ax - c||_2^2 + \alpha\lambda||x||_2 + \frac{(1-\alpha)\lambda}{2}||x||_2^2$$

subject the constraints $Ax = c$, penalty λ and penalty weight α. Also ADMM parameters must be chosen, i.e. step size ρ, tolerances and maximum of iterations allowed. The algorithm also returns the diagnostic variables ϵ_{primal}, ϵ_{dual}, r_{primal} and r_{dual}.

Algorithm 1 ADMM optimized Elastic Net

```
function ELASTIC NET (Method, $A$, $b$, $\lambda$, $\rho_0$, $\alpha$, $\epsilon_{Abs}$, $\epsilon_{Rel}$, MaxIter)

    Choose method, ADMM ($\rho_0 = \rho_1$) or Dual Decomposition ($\rho_1 = 0$)

    Initialization for:
     $k$, $x^{(0)}$, $z^{(0)}$, $y^{(0)}$ and STOP
    Precomutaitons:
     $A^TA$, $A^Tb$, $(A^TA + \mathbf{I}\rho_1)^{-1}$, p and n

    while Stop == $False$ and $k \leq$ MaxIter do

        $x^{(k+1)} = (A^TA + \mathbf{I}\rho_1)^{-1}(A^Tb + \rho_1 z^k - y^k))$
        $z_i^{(k+1)} = \left(\frac{\rho_1 x_i + y_i - \lambda_1}{\lambda_2 + \rho_1}\right)_+ - \left(\frac{-\rho_1 x_i - y_i - \lambda_1}{\lambda_2 + \rho_1}\right)_+$
        $y^{(k+1)} = y^{(k)} + \rho_0(x^{(k+1)} - z^{(k+1)})$

        $r_{Primal} = \epsilon_{Abs}\sqrt{p} + \ \epsilon_{Rel}\ \max\left(||x^{(k+1)}||_2^2,\ ||z^{(k+1)}||_2^2\right)$
        $r_{Dual} \quad = \epsilon_{Abs}\sqrt{n} + \ \epsilon_{Rel}\ ||y^{(k+1)}||_2^2$

        $\epsilon_{primal} \ = ||x^{(k+1)} - z^{(k+1)}||_2^2$
        $\epsilon_{dual} \quad = ||\rho_0(z^{(k+1)} - z^{(k)})||_2^2$

        if $r_{Primal} \ \leq \ \epsilon_{Primal}$ and $r_{Dual} \ \leq \ \epsilon_{Dual}$
            Stop == $True$
        end if

        k=k+1
    end while
return $z, r_{Primal}, r_{Dual}, \epsilon_{primal}, \epsilon_{Dual}$

end function
```

2.2 Parameter experimentation of ADMM optimized Elastic Net in Python using cultivation data

In this section an experiment of Lasso and Elastic net will be conducted using ADMM optimization. Let A denote the gathered measurements from the field trials used in *"Late Blight Prediction and Analysis"* [1] and b profit per hectare. By considering different combinations of the inputs:

Penalty parameter λ

Mixing parameter α

Step size parameter ρ

Absolute stopping criterion $\epsilon_{Absolute}$

Relative stopping criterion $\epsilon_{Relative}$

the experiment will regard the effects in amount of estimated parameters, number of iterations primal primal- respectively dual residuals.

The *Dual Ascent* method has an assumption regarding strict-convexity. Violating this using the algorithm, x converges to $-\infty$. In spite allowing large errors by setting high tolerances did not fix this issue, neither did different choices of ρ and λ.

2.2.1 The effect of the Elastic Net parameters α and λ

The initial scenario

To get a feeling of how λ affect the results a comparison using both *Lasso* and *Elastic Net* for varying λ will be made. Initially evaluating *Elastic Net* with $\lambda = 20$, $\alpha = 0.5$, $\rho = 1$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-2}$ respectively the *Lasso* (where $\alpha = 1$).

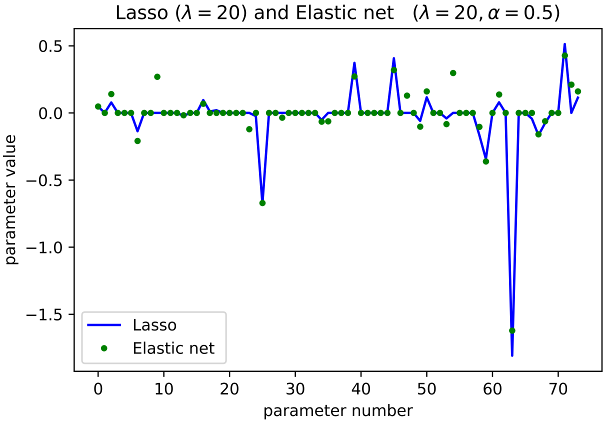
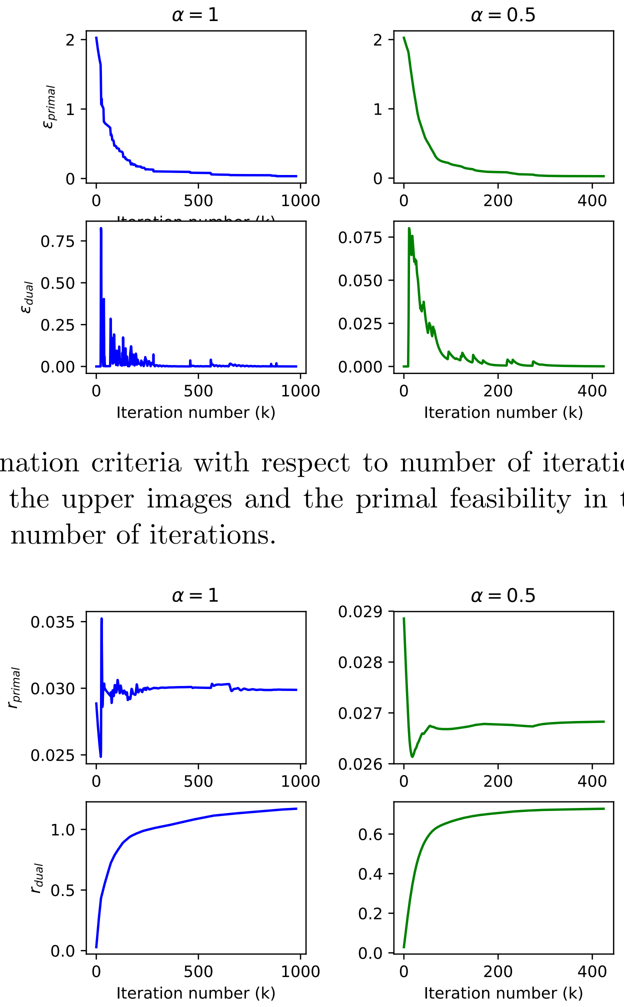


Figure 2.1: Parameters estimated using Elastic net ($\alpha = 0.5$) and Lasso with $\lambda = 20$, $\alpha = 1$, $\rho = 1$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-2}$ as inputs. Further information of the example is given in Section (2.1).

Naturally the estimates lies fairly close to each other as can be seen in Figure (2.1). Assume we have both a *Lasso* and an *Elastic Net* estimate:

$$x^{(Lasso)} = [x_1, 0, x_3, 0, x_5]^T, \qquad x^{(Enet)} = [x_1, x_2, x_3, x_3, 0]^T.$$

Then the maximal number of common active parameters is 3. The actual common active parameters are x_1 and x_2 i.e. 2. Similarly it was found that in this case, Elastic net allows more active parameters i.e. 30 against Lasso which has 24. Generally they agree which parameters that is set to zero, for given inputs they disagree in 4 of 24 possible cases.



(a) Termination criteria with respect to number of iterations. Dual feasibility in the upper images and the primal feasibility in the lower with respect to number of iterations.

(b) Primal- and dual residual with respect to number of iterations.

Figure 2.2: Diagnostics of given example using $\lambda = 20$, $\alpha = 0.5$, $\rho = 1$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-2}$. All images to the right uses $\alpha = 0.5$ and figures to the left $\alpha = 1$.

The first ADMM iterations has the most impact on residuals and stopping criteria see Figure (2.2). After approximately 250 iterations the primal residual ϵ_{primal} barely changes. The dual residual improvement ratio is also slight but proceeds slowly after 250 iterations.

Comparing the number of iterations for Lasso ($\alpha = 1$) respectively Elastic net ($\alpha = 0.5$), it can be seen in Figure (2.2) that Elastic net is faster. Looking at the dual feasibility criterion they do have a similar shape. With respect to the primal feasibility Lasso have more jumping tendencies compared to Elastic net which is more smooth, see Figure (2.2, a, b).

The second scenario

Increasing the penalty parameter to $\lambda = 250$, keeping the other parameters ($\alpha = 0.5$, $\rho = 1$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-2}$) the amount of parameters decreases to 16 in Elastic Net and 10 using Lasso. The methods do not disagree in number of non-zero parameters.

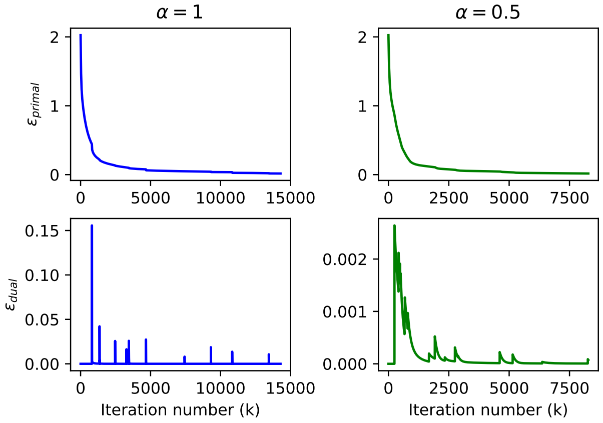


Figure 2.3: ϵ_{primal} and ϵ_{dual} with respect to iteration number using ADMM optimized Elastic net with $\lambda = 250$, $\alpha = 0.5$, $\rho = 1$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-2}$ respectively Lasso where $\alpha = 1$ (Lasso).

Comparing Figure (2.3) to (2.2, a) its notable that ϵ_{primal} is similar in shape, whereas ϵ_{dual} tends to be even less ”smooth” using *Lasso*.

Effect of varying λ and α

Consider fixed $\rho = 50$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-3}$, evaluating *Elastic Net* for $\alpha = 0.02k$ with $k = 0, \ldots, 50$ and $\lambda = 0, 4, 8, \ldots, 400$. Plotting the number of estimated parameters the following image is obtained:

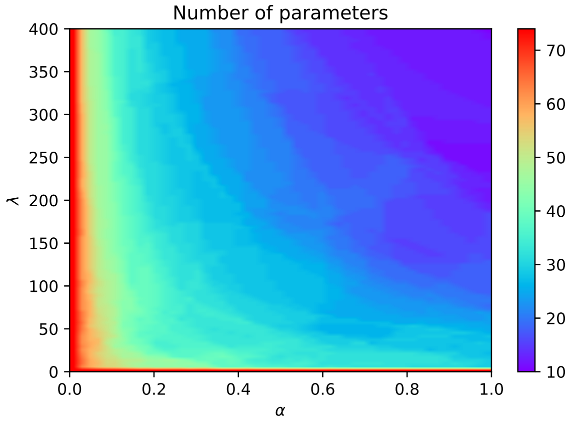


Figure 2.4: Number of estimated *Elastic Net* parameters using $\rho = 50$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-3}$, for $\alpha = 0.02k$ with $k = 0, \ldots, 50$ and $\lambda = 0, 4, 8, \ldots, 400$.

Figure (2.4) confirms that generally increasing the penalty λ results in fewer parameters. Similarly higher weight of the l_1-norm, α, results in more parameters are set to zero. When either λ or α is close enough to zero all parameters are included in the estimation. Note that in Ridge regression ($\alpha = 0$) the solution always includes all parameters. Similarly if $\lambda = 0$ it is regular least squares estimation.

The number of iterations until convergence with respect to α and λ can be illustrated as:

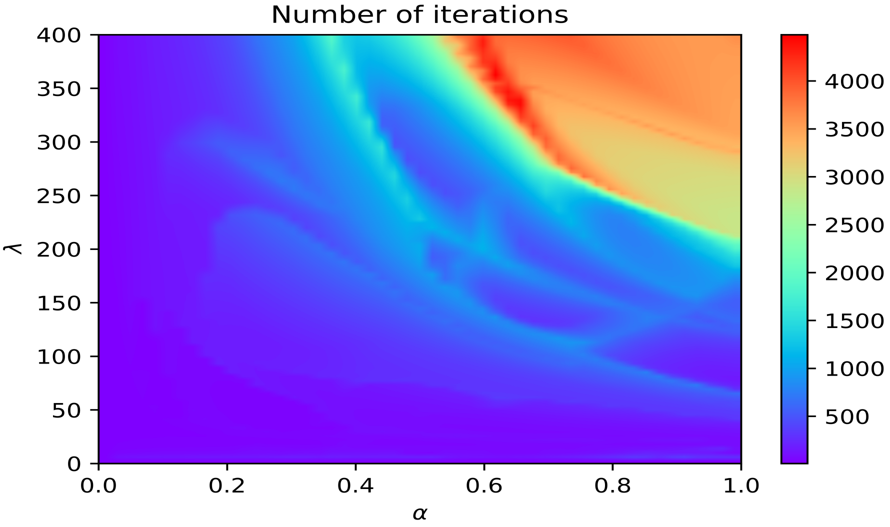


Figure 2.5: Number of *Elastic Net* ADMM iterations with respect to λ and α where $\alpha = 0.02k$, $k = 0, \ldots, 50$ and $\lambda = 0, 4, 8, \ldots, 400$ with fixed $\rho = 50$, $\epsilon_{Absolute} = 10^{-4}$ and $\epsilon_{Relative} = 10^{-3}$.

Figure (2.5) reminds an inversion of Figure (2.4). If either λ or α is large whilst the other is not, the algorithm takes relatively few iterations. If both λ and α are relatively large ($\alpha > 0.8, \lambda > 250$) the amount of iterations increases significantly. Some combinations of λ and α also tends to be faster to solve. Example wise compare $\alpha = 0.4, \lambda = 350$ and $\alpha = 0.45, \lambda = 350$ in Figure (2.5), then it can be seen that the number of iterations locally decrease. This can be interpreted as using more parameters to estimate the objective is faster. If only a few parameters are wanted, then there are more combinations of active parameters to consider.

2.2.2 The effect of ρ and its relation to λ

When deriving the x and z updates one may note that it is possible to write the objective function in several ways. Particularly its possible to introduce the l_2-penalty either in terms of z, x or a combination. Using the latter it is one obtain the updates[2]:

$$x^k = (A^TA + (\rho + \lambda_2/2)\mathbf{I})^{-1}(A^Tc + \rho(z^k - u^k))$$

$$z^k = \frac{(\rho(x+u) - \lambda_1)_+}{\rho + \lambda_2/2} + \frac{(-\rho(x+u) - \lambda_1)_+}{\rho + \lambda_2/2}$$

Using this formulation it is clear that $\lambda_2 = \lambda(1-\alpha) = \lambda_1 \dfrac{1-\alpha}{\alpha}$ is related to ρ.

Evaluating for $\rho = [1, 3, \ldots, 201]$, $\lambda = [1, 1.5, \ldots, 50]$, with fixed $\epsilon_{Absolute} = 10\epsilon_{Relative} = 0.0001$ and $\alpha = 0.5$. Initially plotting the amount of parameters we obtain:

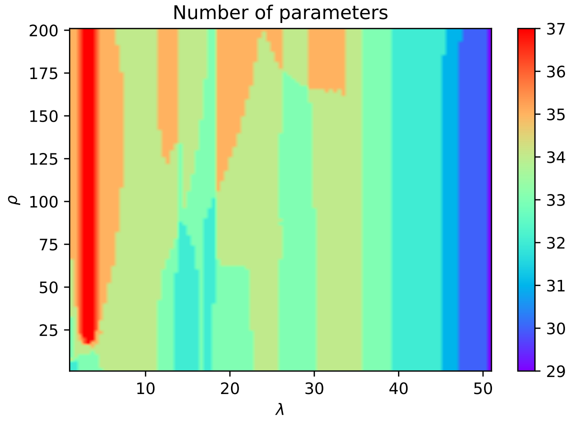
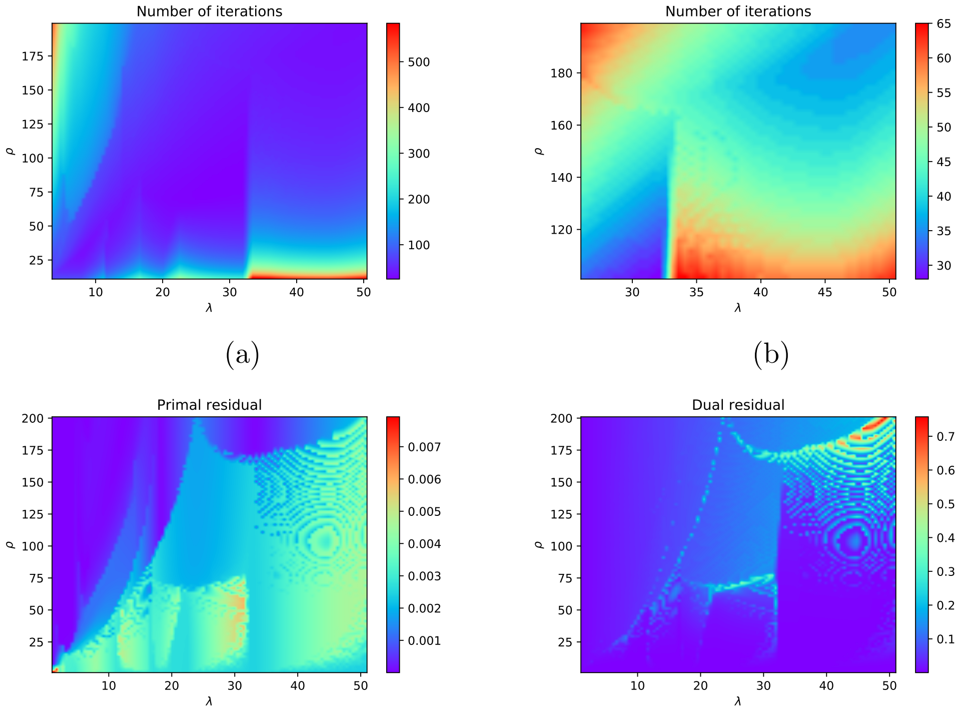


Figure 2.6: Number of *Elastic Net* ADMM parameters with respect to for $\rho = [1, 3, \ldots, 201]$, $\lambda = [1, 1.5, \ldots, 51]$, fixed $\epsilon_{Absolute} = 10\epsilon_{Relative} = 0.0001$ and $\alpha = 0.5$.

It can be seen in Figure (2.6) that as ρ decrease the number of active parameters increase. As λ increase generally less parameters are included in the estimated model. If ρ is much larger than λ it tends to affect the number of parameters heavily.

[2] To get an equivalent expression λ_2 is compensated by 1/2.

Proceeding with the same example plotting the number of iterations and residuals:



(a)

(b)

(c)

(d)

Figure 2.7: Number of *Elastic Net* ADMM iterations with respect to for (a) $\rho = [9, 11, \ldots, 201]$, $\lambda = [2, 2.5, \ldots, 51]$, and (b) $\rho = [101, 103, \ldots, 201]$, $\lambda = [25, 25.5, \ldots, 50]$, fixed $\epsilon_{Absolute} = 10\epsilon_{Relative} = 0.0001$ and $\alpha = 0.5$.

Generally choosing either ρ or λ small relative the other, ADMM tend to be relatively slow, see Figure (2.7.a). If ρ is well chosen, λ does not affect the number of iterations as much. Overall the number of iterations increase together with both ρ and λ, see Figure (2.7, b).

Locally the amount of iterations does not necessarily follow this pattern, see Figure (2.7.b). As *Elastic Net* converges to different solutions depending on the penalty, it is reasonable that also convergence ratio differ. Also it is important to choose an efficient step size in order to improve the convergence. This can be related to the number of parameters included in the estimate, compare to Figure (2.6). If ρ is small, i.e. consider $\rho < \lambda/10$ the dual residual is very small, see Figure (2.7, d). Similarly if $\lambda > \rho/2$ the primal residual is very small compared to the dual residual, see Figure (2.7,c, d).

Looking at Figure (2.7) a suspicion arise regarding whether one should choose ρ depending on the choice of λ[3]. Naively trying a few combinations of λ and ρ for $\rho = 1, 2, \ldots, 200$ and different α. Then summarizing the total number of iterations for each combination we get the following scheme:

α	$\lambda = \rho$	$\lambda = \rho/2$	$\lambda = \rho/4$	$\lambda = \rho/8$	$\lambda = \rho/16$	$\lambda = \rho/32$
1.00	+	+	0	+	+	+
0.90	+	+	0	+	+	+
0.50	+	+	0	0	0	+
0.25	+	+	0	+	+	+
0.10	+	+	0	+	+	+
0.00	0	+	+	+	+	+

Table 2.1: A scheme for different linear combinations of ρ and λ with varying α. If the total number of iterations for $\rho = 1, 2, \ldots, 200$ is larger than one of its left or right neighbors it is denoted by +, if it is less than both its neighbors it is denoted as 0.

From Table (2.1) it can be seen that if $\alpha = 0$ then $\lambda = \rho$ is the best considered choice in terms of iterations. Investigating for $\alpha < 0.05$ resulted in $\lambda = \rho$ also was the best choice. Recall that for very small α all parameters are included in the estimate, in these cases it seems to be preferable with $\rho \leq \lambda$. Else $\rho = 4\lambda$ proved to be the most efficient choice. Similarly it was investigated using $\lambda_2 = \lambda(1 - \alpha)/2$ as reference variable (instead of λ) to ρ. This resulted in larger amount of ADMM iterations for all tested combinations compared to using λ as reference.

[3] Note that we assume that the choice of λ is fixed in real life modelling and ρ is adjusted in the algorithm.

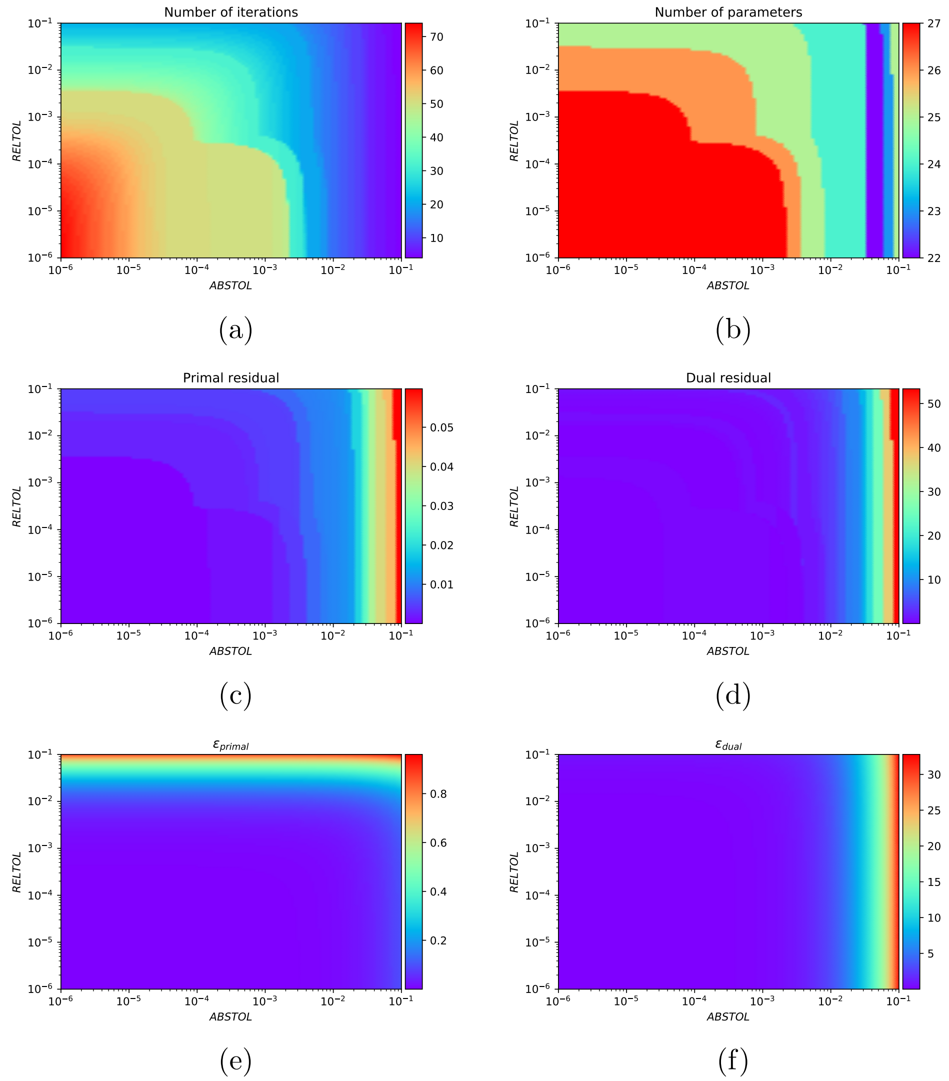
2.2.3 Absolute and Relative tolerances

Fixating the following parameters: $\lambda = 100$, $\rho = 400$, $\alpha = 0.5$ and $MaxIterations = 1000000$ (to assure convergence), we investigate how the absolute- ($ABSTOL$) and relative tolerance ($RELTOL$) affect the results. We consider $ABSTOL$ and $RELTOL$ distributed between 10^{-1} and 10^{-6} with 100 elements. The diagnostics are plotted on the next page.

Studying Figure (2.8) several observations can be made. In Figure (2.8, a) we note that the number of iterations increase using lower tolerances and is slightly more affected by the absolute tolerance. Looking at Figure (2.8, b) it can be seen that for lower tolerances the number of parameters is affected similarly by both tolerances. If ABSTOL is large the number of parameters is mainly affected by the absolute tolerance and seem to have local minima around $10^{-1.6}$. Worth noting is the shapes of amount of iterations and parameters which are very similar.

Both residuals are mostly affected by the absolute tolerance, however there is also a slight effect of the relative tolerance as well. Comparing the residuals to their acceptance criterion, Figure (2.8, e ,d), it is notable that ϵ_{primal} is heavily dependant on the relative tolerance where as ϵ_{dual} is mostly affected by the absolute tolerance. Also notable is where the dual residual is close to convergence with respect to absolute tolerance the number of parameters has past the local minima. Also where the number of iterations start to increase notably the primal residual is close to convergence.

The objective function value decrease as more accuracy is wanted. The absolute tolerance affect it slightly more than the relative. These diagnostics can be used to conclude that the algorithm works as expected, i.e. higher accuracy gives lower objective function value.



(a)

(b)

(c)

(d)

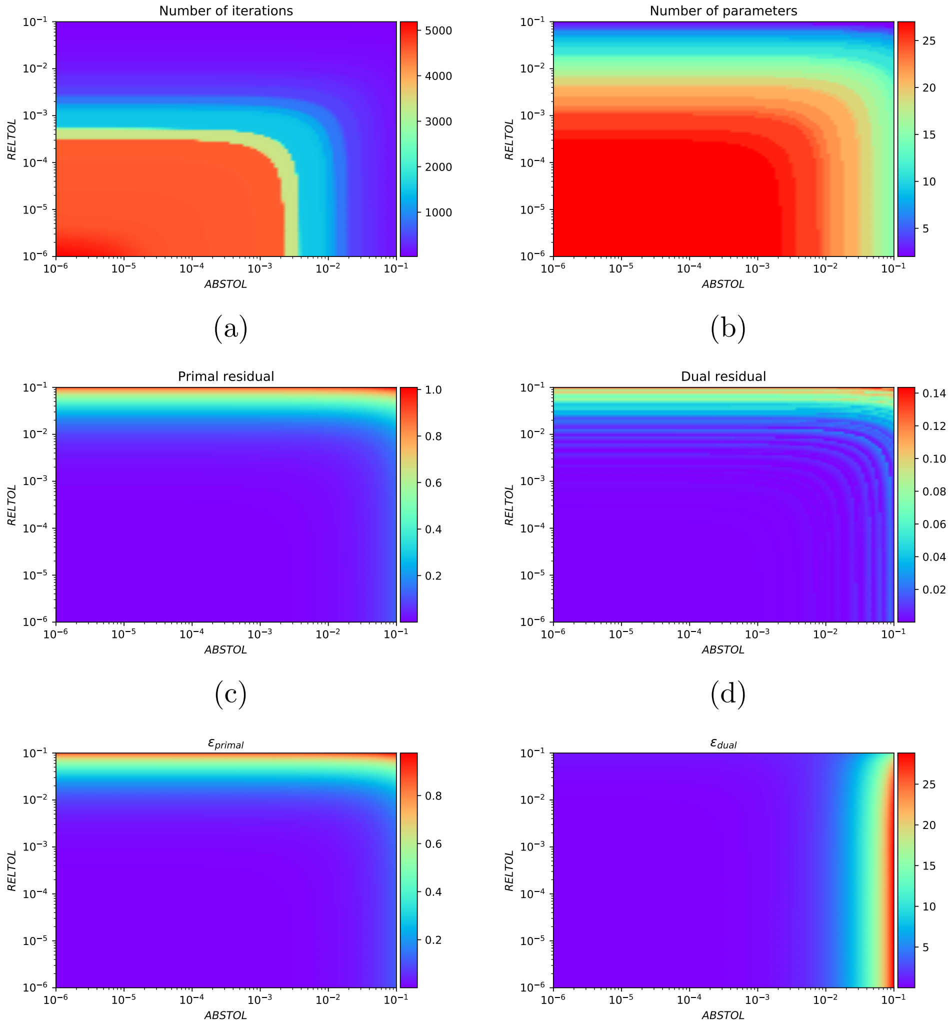
(e)

(f)

Figure 2.8: Diagnostics using fixed $\lambda = 100$, $\rho = 400$, $\alpha = 0.5$ and $MaxIterations = 1000000$ while testing for 100 equidistant points from 10^{-6} and 0.1 for both $ABSTOL$ and $RELTOL$. (a) shows the number of iteration until convergence, (b) number of estimated parameters, (c) ϵ_{primal}, (d) ϵ_{dual}, (e) r_{primal} and (f) r_{dual} at convergence.

A similar analysis was made on a few different combinations of ρ and λ. Initially using $\lambda = 100$ and $\rho = 4$, it was found that the number of parameters increased with respect to both tolerances, see Figure (2.9). The amount of iterations, objective function value, r_{primal}, r_{dual} and ϵ_{primal} was mainly affected by the absolute tolerance. ϵ_{dual} was more affected by the relative tolerance. Notable was that the dual residual has non-smooth tendencies. As in previous example the amount of iterations and number of parameters has related patterns, see Figure (2.9, a, b). The dual residual non-smooth tendencies also seem to be related to change in the number of parameters.

Also $\lambda = 100$ and $\rho = 4000$ was considered, see Figure (2.10). In this example the number of parameters quickly stabilize at 27 and only the absolute tolerance seem to affect. The number of iterations as a similar shape as in previous example (Figure 2.9) but is mostly affected by the relative tolerance instead of the absolute tolerance. Notable is the combined shape of amount of parameters in this and previous example would relate to the first, see Figure (2.8, 2.9, 2.10). The dual residual is mainly affected by the absolute tolerance in comparison to the other examples where the relative tolerance had more influence. Studying the accepted primal residual one can see that it has jump tendencies with respect to the absolute tolerance in particular when it has a low tolerance.



(a)

(b)

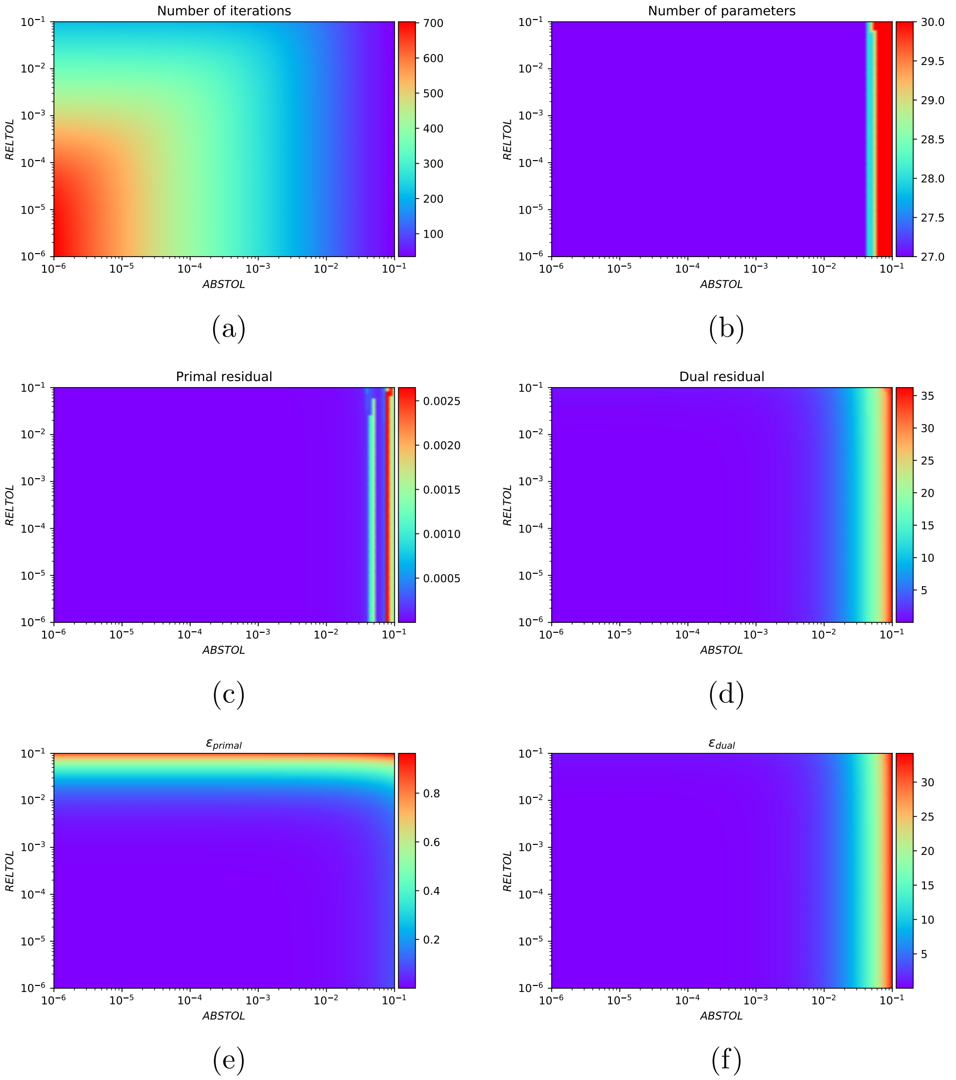
(c)

(d)

(e)

(f)

Figure 2.9: Diagnostics using fixed $\lambda = 100$, $\rho = 4$, $\alpha = 0.5$ and $MaxIterations = 1000000$ while testing for 100 equidistant points from 10^{-6} and 0.1 for both $ABSTOL$ and $RELTOL$. (a) shows the number of iteration until convergence, (b) number of estimated parameters, (c) ϵ_{primal}, (d) ϵ_{dual}, (e) r_{primal} and (f) r_{dual} at convergence.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.10: Diagnostics using fixed $\lambda = 100$, $\rho = 4000$, $\alpha = 0.5$ and $MaxIterations = 1000000$ while testing for 100 equidistant points from 10^{-6} and 0.1 for both $ABSTOL$ and $RELTOL$. (a) shows the number of iteration until convergence, (b) number of estimated parameters, (c) ϵ_{primal}, (d) ϵ_{dual}, (e) r_{primal} and (f) r_{dual} at convergence.

Chapter 3

Conclusion

The *Alternating Direction Method of Multipliers* is fairly old but still up-to-date. It is very adaptable but the convergence ratio may lack compared to other algorithms. ADMM takes the best from its precursors, the decomposability from *Dual Ascent* and *Method of Multipliers* convergence properties. This paper is an experiment using data from a cultivation trial, investigating parameter effects of ADMM optimization and *Elastic Net*.

Dual Decomposition is the most basic of the considered optimization methods. It is based optimize the primal problem by solving the dual problem iteratively. Unfortunately it works only under rather strict assumptions and is heavily dependant on the step size choice [3]. *Method of Multipliers* has better convergence properties. But due to the augmentation its rarely decomposable. Example wise the following issue appears in an *Elastic Net* setting:

$$\frac{(A^TA)^{-1}\partial(\lambda||x||_1)}{1+\rho} + x = (A^TA)^{-1}\left(Ac - \frac{y^TA + \lambda(1-\alpha)}{1+\rho}\right).$$

Due to no efficient solution was found, this method was not further investigated.

Between iterations it was found that *Lasso* had very non-smooth updates, which follows from the l_1-penalty. When a mixture of l_1- and l_2-penalty was considered (*Elastic Net*) the updates became smoother. As expected *Elastic Net* included more parameters and proved to take less iterations compared to *Lasso*.

Investigating the effect of arbitrary parameters in ADMM and *Elastic Net* it was found that increasing the penalty λ decreased the number of parameters. If λ is large enough only one parameter will remain. Similarly adjusting the weight between the one- and two- norm, α, it was confirmed that as α decreased the amount of parameters increased (i.e. include relatively more l_2-penalty).

For increasing ρ ADMM tend to include more parameters in the result, however the effect depends on the relation between ρ and λ. As λ grow *Elastic Net* put a heavier penalty on the parameters reducing the amount of parameters in the estimated model. Notably is that in Ridge regression, i.e. only including the l_2-penalty, the amount of parameters remain unaffected. Important is to emphasise that if ρ is much larger than λ it affect the result which is not preferable. However if ρ is to small the amount of iterations may heavily increase.

Studying the number of ADMM iterations it was found that there was a relation in efficiency with respect to ρ and λ. Choosing either large λ and small ρ, or vice versa, ADMM tends to need more iterations. Else it preforms relatively well with local variations. For the given scenario $\lambda = \rho/4$ was the most efficient choice of ρ overall.

Lastly relative- and absolute tolerance effect was investigated. Higher accuracy allowed more parameters and demanded more ADMM iterations. The objective function had lower value hence was improved using lower tolerances.

Bibliography

[1] L. Bengtsson, "Late blight prediction and analysis," 2017. Student Paper.

[2] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Transactions of the American mathematical Society*, vol. 82, no. 2, pp. 421–439, 1956.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[4] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, vol. 162, no. 1-2, pp. 165–199, 2017.

[5] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of admm," *arXiv preprint arXiv:1502.02009*, 2015.

[6] Stephen P. Boyd, "Alternating Direction Method of Multipliers." `http://videolectures.net/nipsworkshops2011_boyd_multipliers/`, 2013. Department of Electrical Engineering, Stanford University, Online; accessed 2017-09-11.

[7] P. E. Gill, W. Murray, and M. H. Wright, "Practical optimization," 1981.

[8] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*, vol. 1. Springer Science & Business Media, 2006.

[9] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming.* Kluwer Academic Publishers, 2015.

[10] G. Hall, “Convex and conic optimization,” 2015. Princeton University, Online; accessed 2017-09-27.

[11] H. P. Williams, *Model building in mathematical programming.* John Wiley & Sons, 2013.

[12] N. Parikh, S. Boyd, *et al.*, “Proximal algorithms,” *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[13] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations.* CRC press, 2015.

[14] A. E. Hoerl and R. W. Kennard, “Ridge regression: applications to nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970.

[15] S. Boyd and L. Vandenberghe, “Subgradients,” 2006. Stanford University, Online; accessed 2017-01-16.