TMA4180 Optimization

477789, 491531, 501037

March 24, 2019

Abstract

This report looks into various optimization problems arising in data segmentation and model fitting. Data segmentation is very important in the industry, especially in the field of machine learning and unsupervised learning. An example is marketing companies gaining customer insight by separating their customer databases into segments of high-opportunity and low-opportunity customers, and focusing on the high-opportunity segment to increase profit. In this case data segmentation can be very profitable for the company. Another example is separating medical patients into high- and low-risk groups for some disease. In this case data segmentation can be life saving. Specifically, this report looks at separating data with hypersurfaces such as ellipsoids and hyperboloids according to labels. The data is labeled according to two different models, and we compare performance of the implemented algorithms on these two models. We consider both constrained and unconstrained settings, and solve the problems by implementing suitable optimization algorithms for both models in each setting. We also expand the constrained setting to take three different labels into account. For the unconstrained setting we implement gradient descent- and the BFGS method, whereas for the constrained setting we implement the augmented Lagrangian method. The unconstrained algorithms work well for both models in the test cases we considered, with some differences regarding convergence. In the constrained binary classification setting, the augmented Lagrangian objective function with gradient descent performs well when tuned correctly. In the three label setting, the performance of this algorithm is heavily dependent on the initial guess.

Introduction

In this report we present theory and numerical results to various optimization problems, both unconstrained and constrained, occurring in data segmentation. We will consider two types of segmentations. The first we will consider is a binary classification with two labels, which we solve in an unconstrained and constrained manner. The second type of segmentation we consider is a case in which the data can have three different labels, which we solve in a constrained manner.

In the unconstrained setting we implement the gradient descent method with backtracking line-search as well as the BFGS method with bisection line-seach satisfying the Wolfe-conditions. In the constrained settings we implement the augmented Lagrangian method.

In the Main part section we define the optimization problems and discuss the different algorithms we have implemented. Further on, we present the numerical results on different test cases in the section Numerical experiments. Finally, we summarize our finding in the Summary section.

Main part

Theory

Unconstrained setting

The two sets corresponding to the different models are defined as follows.

$$S_{A,c} = \{x \in \mathbb{R}^d : (x-c)^T A(x-c) \leq 1\} \tag{1}$$

and

$$E_{A,b} = \{x \in \mathbb{R}^d : x^T Ax - x^T b \leq 1\}. \tag{2}$$

For the first representation, $S_{A,c}$, the objective function to fit the datapoints is

$$f_1(A,c) = \sum_{w_i=a} \max\{r_i(A,c),0\}^2 + \sum_{w_i=b} \min\{r_i(A,c),0\}^2 \tag{3}$$

where

$$r_i(A,c) = (z_i-c)^T A(z_i-c) - 1.$$

This function penalizes every point labeled as *inside* which is outside and visa-versa. The function f_1 is an element of C^1 but not necesarilly of C^2. It is an element of C^1 since all its first order derivatives are defined and continuous. They are equal to

$$\frac{\partial f_1}{\partial A} = \sum_{w_i=a,\ r_i\geq 0} 2r_i(A,c)(z_i-c)(z_i-c)^T + \sum_{w_i=b,\ r_i\leq 0} 2r_i(A,c)(z_i-c)(z_i-c)^T$$

and

$$\frac{\partial f_1}{\partial c} = \sum_{w_i=a,\ r_i\geq 0} 4r_i(A,c)A(z_i-c) + \sum_{w_i=b,\ r_i\leq 0} 4r_i(A,c)A(z_i-c)$$

∇f_1 can jump in function value at the boundary of a domain $\Omega_i = \{A,c,z_i \mid w_i = a,\ r_i \geq 0 \cap\ w_i = b,\ r_i \leq 0\}$. However, every term constituting to this jump approaches zero at the boundary of Ω_i,

$$2r_i(A,c)\frac{\partial r_i}{\partial A} \to 0, \quad A,c \to \partial\Omega_i.$$

Therefore, ∇f_1 is continuous. Since ∇f_1 becomes zero at the boundary of Ω, it is continuous and therfore, $f_1 \in C^1$. The Hessian Δf_1 in general has a jump at the boundary of Ω_i, hence it is not continuous and $f_1 \notin C^2$.

The function f_1 is unlikely to be convex. The function $r_i(A,c)$ describes a parabola in c which can be both negative and positive, taking the squared maximum results in a function with multiple minima, which occur when $r_i(A,c)$ is equal to zero. The function $f_1(A,c)$ is however coercive and lower semi-continuous. The coercivity follows from the limit $r_i(A,c) \to \infty$ if $A,c \to \infty$ and the lower semi-continuity from the continuity of f_1. Therefore, the solution of

$$\min_{A,c\ A\in \mathrm{Sym}_d} f_1(A,c) \tag{4}$$

exists but is not unique.

The function f_2 is the objective function for the second representation $E_{A,b}$,

$$f_2(A,b) = \sum_{w_i=a} \max\{\tilde{r}_i(A,b),0\}^2 + \sum_{w_i=b} \min\{\tilde{r}_i(A,b),0\}^2 \tag{5}$$

where

$$\tilde{r}_i(A,b) = z_i^T A z_i - z_i^T b - 1.$$

The function f_2 is convex. We will prove this in three steps. First, $h = \max\{0,t\}^2$ is convex. The definition of convexity is

$$h(\lambda x + (1-\lambda)y) \leq \lambda h(x) + (1-\lambda)h(y).$$

which we can rewrite as

$$\max\{0, \lambda x + (1-\lambda)y\}^2 \leq \lambda \max\{0,x\}^2 + (1-\lambda)\max\{0,y\}^2,$$

then, we identify two cases which cover all possibilities since x and y can be swapped.

1. If $y > x \geq 0$, then we have
$$\lambda x^2 + (1-\lambda)y^2 \leq \lambda x^2 + (1-\lambda)y^2$$
2. If $y \geq 0 > x$, then we have
$$\max\{0, \lambda x + (1-\lambda)y\}^2 \leq (1-\lambda)y^2$$
Since $y > 0$ and $x < 0$, $\lambda x + (1-\lambda)y < (1-\lambda)y$

A similar argument holds for the squared minimum. Secondly, if $\max\{0,t\}^2$ is convex then $\max\{0,t(u)\}^2$ is convex for any linear mapping $t(u)$, thus also for the mapping $\tilde{r}_i(A,c)$. Lastly, an affine transformation, such as the sum, of a convex function is also convex. Since f_2 is convex, the solution of the problem

$$\min_{A,c \ A\in \mathrm{Sym}_d} f_2 \tag{6}$$

exists and is unique.

The gradient of f_2, ∇f_2 is Lipschitz continuous. The componentes of ∇f_2 for $A, c \in \Omega$ are given by:

$$\frac{\partial f_2}{\partial A} = \sum_{w_i=a,\ \tilde{r}_i \geq 0} 2\tilde{r}_i(A,c) z_i z_i^T + \sum_{w_i=b,\ \tilde{r}_i \leq 0} 2\tilde{r}_i(A,c) z_i z_i^T.$$

and

$$\frac{\partial f_2}{\partial b} = \sum_{w_i=a,\ \tilde{r}_i \geq 0} 2\tilde{r}_i(A,c) z_i + \sum_{w_i=b,\ \tilde{r}_i \leq 0} 2\tilde{r}_i(A,c) z_i$$

The possible jumps in the function ∇f_2 occur at the boundary of $\tilde{\Omega}_i = \{A, c, z_i \mid w_i = a,\ \tilde{r}_i \geq 0 \cap\ w_i = b,\ \tilde{r}_i \leq 0\}$, where $\tilde{r}_i$ becomes zero. Therefore, the jumps have size zero and ∇f_2 is Lipschitz continuous.

Constrained setting

We are still concerned with minimizing the objective functions given by (3) and (5). For more convenient notation, we are denoting

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T = \begin{bmatrix} A_{11} & A_{12} & A_{22} & c_1 & c_2 \end{bmatrix}^T.$$

That is, we are trying to optimize $x \in R^5$, where c_1 and c_2 is the center of the ellipses.

But now, we are adding constraints given by

$$\begin{cases} c_1: & x_1 - \gamma_1 \geq 0 \\ c_2: & \gamma_2 - x_1 \geq 0 \\ c_3: & x_3 - \gamma_1 \geq 0 \\ c_4: & \gamma_2 - x_3 \geq 0 \\ c_5: & \sqrt{(x_1 \cdot x_3)} - \sqrt{\gamma_1^2 + x_2^2} \geq 0 \end{cases} \tag{7}$$

First, we will show that the function, denoted g,

$$g : (x_1, x_2, x_3) \Rightarrow \sqrt{(x_1 x_3)} - \sqrt{\gamma_1^2 + x_2^2}$$

is C^2 and concave. We start with finding all the second derivatives of the function, using standard methods for derivation

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} & \frac{\partial g}{\partial x_3} \end{bmatrix}^T = \begin{bmatrix} \frac{\sqrt{x_3}}{2\sqrt{x_1}} & -\frac{x_2}{\sqrt{\gamma_1^2+x_2^2}} & \frac{\sqrt{x_1}}{2\sqrt{x_3}} \end{bmatrix}^T$$

$$\nabla^2 g = \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2} & \frac{\partial^2 g}{\partial x_1 x_2} & \frac{\partial^2 g}{\partial x_1 x_3} \\ \frac{\partial^2 g}{\partial x_2 x_1} & \frac{\partial^2 g}{\partial x_2^2} & \frac{\partial^2 g}{\partial x_2 x_3} \\ \frac{\partial^2 g}{\partial x_3 x_1} & \frac{\partial^2 g}{\partial x_3 x_2} & \frac{\partial^2 g}{\partial x_3^2} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{x_3}}{4x_1^{\frac{3}{2}}} & 0 & \frac{1}{4\sqrt{x_1 x_3}} \\ 0 & -\frac{\gamma_1^2}{(\gamma_1^2+x_2)^{\frac{3}{2}}} & 0 \\ \frac{1}{4\sqrt{x_1 x_3}} & 0 & -\frac{\sqrt{x_1}}{4x_3^{\frac{3}{2}}} \end{bmatrix} \tag{8}$$

since we require that $x_1 > 0$ and $x_3 > 0$, all second derivatives are well defined. Hence the function is in C^2. In order for the function to be concave, it has to be in C^2 and the Hessian matrix, given by (8), has to be negative semi-definite. To prove this, we are considering the determinants of the k-dimensional sub matrices $\nabla^2 g_k$, for $k \in \{1, 2, 3\}$. In order to be negative semi-definite, $(-1)^k \det(\nabla^2 g_k) \geq 0$. Using standard methods for calculating determinants, we obtain:

$$\mathbf{k = 1:} \qquad (-1)^1 \det(\nabla^2 g_1) = -\det\big(-\frac{\sqrt{x_3}}{4x_1^{\frac{3}{2}}}\big) > 0$$

$$\mathbf{k = 2:} \qquad (-1)^2 \det(\nabla^2 g_2) = \det \begin{bmatrix} -\frac{\sqrt{x_3}}{4x_1^{\frac{3}{2}}} & 0 \\ 0 & -\frac{\gamma_1^2}{(\gamma_1^2+x_2)^{\frac{3}{2}}} \end{bmatrix} > 0$$

$$\mathbf{k = 3:} \qquad (-1)^3 \det(\nabla^2 g_3) = -\det \begin{bmatrix} -\frac{\sqrt{x_3}}{4x_1^{\frac{3}{2}}} & 0 & \frac{1}{4\sqrt{x_1 x_3}} \\ 0 & -\frac{\gamma_1^2}{(\gamma_1^2+x_2)^{\frac{3}{2}}} & 0 \\ \frac{1}{4\sqrt{x_1 x_3}} & 0 & -\frac{\sqrt{x_1}}{4x_3^{\frac{3}{2}}} \end{bmatrix} = 0$$

Hence the Hessian matrix is negative semi-definite, and the function is concave.

Lets define the set described by our constraints as

$$\Omega = \{x \in \mathbb{R}^3 : c_i(x) \geq 0, \quad i \in \mathcal{I}\}$$

the set Ω is convex if

$$\alpha x + (1-\alpha)\hat{x} \in \Omega$$

for all $x, \hat{x} \in \Omega$ and $\alpha \in (0, 1)$. We have previously showed that all of our inequality constraints are concave. Hence we can use the definition of concavity, assuming that $c_i(x), c_i(\hat{x}) \geq 0$ and $\alpha, (1-\alpha) \geq 0$.

$$c_i(\alpha x + (1-\alpha)\hat{x}) \geq \alpha c_i(x) + (1-\alpha) c_i(\hat{x}), \quad \forall x, \hat{x} \in \Omega$$

from our assumptions we obtain

$$c_i(\alpha x + (1-\alpha)\hat{x}) \geq 0$$

which proves that $\alpha x + (1-\alpha)\hat{x} \in \Omega$, and thus Ω is a convex set. Also, the constraints are continuous functions defined on closed intervals, and thus the set is closed.

In constrained optimization, we often define necessary KKT-conditions in the following way. Assume $x^* \in \mathbb{R}^5$ is a local solution of (4) or (6). The linear independence constraint qualification (LICQ) states that the gradients of the active constraints should be linearly independent at x^*. The gradients of c_1 and c_2, as well as c_3 and c_4, are linearly dependent, but since $\gamma_2 > \gamma_1$ by definition, c_1 and c_2 can not be active at the same time. The exact same argument holds for c_3 and c_4. Hence the gradient of the active constraints linearly independent, and the LICQ is satisfied at x^*. Then there exists Lagrange multipliers $\lambda^* \in \mathbb{R}^5$ such that

$$\lambda_i^* > 0 \quad \text{for} \quad i \in \{1, 2, 3, 4, 5\}$$

$$c_i(x) > 0 \quad \text{for} \quad i \in \{1, 2, 3, 4, 5\}$$

$$\lambda_i^* c_i(x) = 0 \quad \text{for} \quad i \in \{1, 2, 3, 4, 5\}$$

$$\nabla f(x^*) = \sum_i \lambda_i^* \nabla c_i(x)$$

In order to decide whether these conditions also are sufficient optimality conditions, we need a convex problem and the Slater's constraint qualification has to be satisfied.

Slater's constraint qualification is satisfied if all the inequality constraints have to be concave and if there exists a point $\hat{x}$ for which $c_i(\hat{x}) > 0$ for $i \in \{1, 2, 3, 4, 5\}$. Previously, we proved that the last constraint is concave, and due to the linearity of the other constraints, it follows that they are concave as well. By choosing the point

$$\hat{x} = \begin{bmatrix} \frac{\gamma_1+\gamma_2}{2} & 0 & \frac{\gamma_1+\gamma_2}{2} & c_1 & c_2 \end{bmatrix}^T,$$

all inequality constraints are greater than zero for all c_1 and c_2. Thus we can conclude that the constraints satisfy Slater's constraint qualification.

Since (3) is not necessarily convex, the KKT-conditions do not have to be sufficient here. We have shown that (5) is a convex function, thus the KKT-conditions are both necessary and sufficient for this problem.

Three label data segmentation

We now expand our problem such that our data points can have three possible labels $w_i \in \{a, b, c\}$. Points with labels a and b belongs to two separate sets R_1 lnd R_2, respectively. Points with label c are not contained by any of the sets We define our sets by

$$R_i = \{\|z - c_i\|^2 \leq \rho^2\}$$

As before, we now define the residuum as

$$r_i(c, \rho) := \|z - c_i\|^2 - \rho^2$$

so we will attempt to minimize the function

$$f_3(c, \rho; d, \sigma) = \sum_{w_i=a} max\{r_i(c, \rho), 0\}^2 + \sum_{w_i \in \{b,c\}} min\{r_i(c, \rho), 0\}^2 + \sum_{w_i=b} max\{r_i(d, \sigma), 0\}^2 + \sum_{w_i \in \{a,c\}} min\{r_i(d, \sigma, 0\}^2 \tag{9}$$

subject to the constraints

$$\begin{cases} c_1 : & \rho \geq 0 \\ c_2 : & \sigma \geq 0 \\ c_3 : & \|c - d\| - \rho + \sigma \geq 0 \end{cases} \tag{10}$$

Lets denote

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T = \begin{bmatrix} c_1 & c_2 & \rho & d_1 & d_2 & \sigma \end{bmatrix}^T$$

to be the solution vector we are trying to optimize, where c_1, d_1 is the x-coordinates of the centers, c_2, d_2 is the y-coordinates, and ρ and σ is the radiuses.

First, we will discuss the convexity of the function. This was analyzed by using the definition of a convex function to consider the convexity of r_i. As r_i is a function of two variables, c and ρ, we have to check if r_i is convex subject to both parameters. For it to be convex in c,

$$r_i(\alpha c + (1 - \alpha)\hat{c}, \rho) \leq \alpha f(c, \rho) + (1 - \alpha) f(\hat{c}, \rho)$$

$$\|z_i - (\alpha c + (1 - \alpha)\hat{c}\|^2 - \rho^2 \leq \alpha(\|z_i - c\|^2 - \rho^2) + (1 - \alpha)(\|z_i - \hat{c}\|^2 - \rho^2)$$

here, the ρ terms are equal on both sides, and by using the definition of the Euclidean norm we obtain

$$(z_{i1} - \alpha c_1 - (i - \alpha)\hat{c_1})^2 + (z_{i2} - \alpha c_2 - (i - \alpha)\hat{c_2})^2 \leq \alpha \left((z_{i1} - c_1)^2 + (z_{i2} - c_2)^2\right) + (1 - \alpha) \left((z_{i1} - \hat{c_1})^2 + (z_{i2} - \hat{c_2})^2\right)$$

by solving the parantheses and getting rid of all the terms which level out, we obtain the following inequality

$$\alpha^2 c_1^2 + 2\alpha(1 - \alpha)c_1\hat{c_1} + \alpha^2 c_2^2 + (1 - \alpha)^2 \hat{c_1}^2 + 2\alpha(1 - \alpha)c_2\hat{c_2} + (1 - \alpha)^2 \hat{c_2}^2 \leq \alpha c_1^2 + \alpha c_2^2 + (1 - \alpha)\hat{c_1}^2 + (1 - \alpha)\hat{c_1}^2$$

by moving all the terms to the right side of the inequality and merging the terms of the same variable, we get

$$0 \leq \alpha(1 - \alpha)c_1^2 + \alpha(1 - \alpha)c_2^2 + \alpha(1 - \alpha)\hat{c_1}^2 + \alpha(1 - \alpha)\hat{c_2}^2 - 2\alpha(1 - \alpha)c_1\hat{c_1} - 2\alpha(1 - \alpha)c_2\hat{c_2}$$

when dividing by $\alpha(1 - \alpha)$ we can rewrite the inequality as

$$0 \leq (c_1 - \hat{c_1})^2 + (c_2 - \hat{c_2})^2$$

which clearly is true. Hence we conclude that the r_i is convex subject to c.

Similarly, for r_i to be convex subject to ρ, the following inequality has to hold:

$$r_i(c, \alpha\rho + (1 - \alpha)\hat{\rho}) \leq \alpha f(c, \rho) + (1 - \alpha) f(c, \hat{\rho})$$

which yields

$$\|z_i - c\|^2 - (\alpha\rho + (1-\alpha)\hat{\rho})^2 \leq \alpha(\|z_i - c\|^2 - \rho^2) + (1-\alpha)(\|z_i - c\|^2 - \hat{\rho}^2)$$

again, the terms independent of ρ levels out, and we get

$$-(\alpha^2\rho^2 + 2\alpha(1-\alpha)\rho\hat{\rho} + (1-\alpha)^2\hat{\rho}^2) \leq -\alpha\rho^2 - (1-\alpha)\hat{\rho}^2$$

by multiplying with -1 and flipping the inequality sign, moving all terms to the right, and merging terms, we get

$$0 \geq \alpha(1-\alpha)\rho^2 + \alpha(1-\alpha)\hat{\rho}^2 - 2\alpha(1-\alpha)\rho\hat{\rho}$$

which gives

$$0 \geq (\rho - \hat{\rho})^2$$

which clearly is not true. Hence we conclude that r_i is not convex subject to ρ. Thus we conclude that the function r_i is not convex.

The function f_3 is a pretty complex function, which is difficult to analyse analytically. From (9), we see that the function f_3 is basically defined as a sum of the functions r_i squared. Since we have shown that the function r_i is not convex, it indicates that the function f_3 might not be convex either. Consequently we can not necessarily guarantee that the minimum of f_3 has a unique solution.

Lets denote the set given by the constraints in (10), as

$$\Psi = \{x \in R^6 : c_i(x) \geq 0, \quad i \in \mathcal{I}\}$$

the first two constraints given by (10), are both concave and convex due to their linearity. However, the last constraint is a function of the Euclidean norm, which we can easily prove is a convex function. Let V be a vectorspace, and define a norm $\|\cdot\| : V \Rightarrow \mathbb{R}$. Using the definition of norms, the triangle inequality and the scaleability of norms, then $\forall x, \hat{x} \in V$, and a constant $\alpha \in (0,1)$, we obtain

$$\|\alpha x + (1-\alpha)\hat{x}\| \leq \alpha\|x\| + (1-\alpha)\|\hat{x}\|$$

which is exactly the definition of a convex function. Hence we conclude that all norms are convex, and the Euclidean norm is thus a convex function. The linear terms do not change the convexity of the constraint function, and the last constraint is hence convex. Using similar argumentation as for the previous constrained setting, we can in this case conclude that the set described by the constraints is not convex.

Based on the constraints in (10), We can formulate KKT-conditions also for this problem. Assume a local solution $x^* \in R^6$, here the gradients of all constraints are clearly linearly independent, so the LICQ holds at x^*. Then there exists Lagrange parameters $^* \in R^3$ such that

$$\lambda_i^* > 0 \quad \text{for} \quad i \in \{1,2,3\}$$

$$c_i(x) > 0 \quad \text{for} \quad i \in \{1,2,3\}$$

$$\lambda_i^* c_i(x) = 0 \quad \text{for} \quad i \in \{1,2,3\}$$

$$\nabla f(x^*) = \sum_i \lambda_i^* \nabla c_i(x)$$

since not all constraints are concave, the Slaters inequality condition is not satisfied. Hence the KKT-conditions is necessary but not sufficient.

Optimization algorithms

Unconstrained setting

We are concerned with the following problems

$$\min_{A \in \mathrm{Sym}_2, c \in \mathbb{R}^2} f_1(A,c) \quad \text{or} \quad \min_{A \in \mathrm{Sym}_2, b \in \mathbb{R}^2} f_2(A,b) \tag{11}$$

If we restrict ourselves to only deal with real,symmetric matrices in $\mathbb{R}^{2\times 2}$ we can solve problem (11) by means of unconstrained optimization algorithms.
To solve the unconstrained problems we implemented two different algorithms to minimize the objective functions f_1 and f_2 for model 1 and model 2. We chose to focus on gradient descent and the BFGS method.

Gradient descent is a simple, first-order iterative optimization algorithm that seeks to minimize a function f by taking steps in the direction of the negative gradient at the current point. Thus, the search direction at point

x_k is $p_k = -\nabla f_k$.

The BFGS method belongs to a set of methods known as the quasi-Newton methods. The quasi-Newton methods use the Newton search direction, $p_k = -(\nabla^2 f_k)^{-1}\nabla f_k$, with an approximation of the hessian. This makes the methods more sophisticated than the gradient descent as they include more information of the objective function. The BFGS method approximate the hessian $\nabla^2 f_k$ at each iteration by B_k found by the secant equation

$$B_{k+1}s_k = y_k,$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$.
The BFGS step is calculated as

$$H_{k+1} = (I - \rho_k s_k y_k^T)H_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T,$$

where $\rho_k = (y_k^T s_k)^{-1}$ and $H_k = B_k^{-1}$. Substituting this into the newton search direction yields the BFGS search direction $p_k = -H_k\nabla f_k$. To ensure that the search direction at each iterate is a descent direction, we implemented the BFGS method with a bisection line-search with $\alpha_0 = 1$ satisfying the Wolfe-conditions as described below.

Constrained setting

Binary classification In the case that the matrix A in the sets $S_{A,c}$ and $E_{A,c}$ is not positive definite, the resulting hypersurface is a hyperbola. In this section we will discuss the augmented Lagrangian method to force the solution A^* to be positive definite. The constraints which enforce this are

$$\gamma_1 \leq A_{11} \leq \gamma_2$$
$$\gamma_1 \leq A_{22} \leq \gamma_2$$
$$(A_{11}A_{22})^{\frac{1}{2}} \geq (\gamma_1^2 + A_{12}^2)^{\frac{1}{2}}$$

for $0 < \gamma_1 < \gamma_2 < \infty$. The augmented Lagrangian method uses three constraint functions,

$$c_1(x) = \max\{0, (A_{11} - \gamma_1)(A_{11} - \gamma_2)\},$$
$$c_2(x) = \max\{0, (A_{22} - \gamma_1)(A_{22} - \gamma_2)\}$$

and

$$c_3(x) = \max\{0, \sqrt{\gamma_1^2 + A_{12}^2} - \sqrt{A_{11}A_{22}}\}$$

to modify the objective function

$$f_{\text{augL}} = f_m(x) - \sum_{i=1}^{3}\lambda_i c_i(x) + \frac{\mu^2}{2}\sum_{i=1}^{3} c_i(x)^2$$

where f_m can be equal to f_1 or f_2 depending on the model chosen. The function f_{augL} is equal to f_m as long as the iterates satisfy the constraints. Around the domain $\Gamma = \{x \mid c_1(x) = 0 \cap c_2(x) = 0 \cap c_3(x) = 0\}$, the function value of f_{augL} increase steeply. The steepness of the 'wall' on the boundary is essential for the proper working of the algorithm. Figure 4 illustrates these rising boundaries.

Three label data segmentation The constraints in force the circles to be well defined, with a radius > 0 and non-overlapping. The following set defines a circle i

$$R_i = \{||z - c_i||^2 \leq \rho_i^2\}.$$

The constraints in the case of two circles are thus

$$\rho, \sigma \geq 0$$
$$||c - d|| \geq \rho + \sigma$$

To construct the augmented Lagrangian function

$$f_{augL} = f_m(x) - \sum_{i=1}^{3}\lambda_i c_i(x) + \frac{\mu^2}{2}\sum_{i=1}^{3} c_i(x)^2$$

we defined the following constraint functions

$$c_1(x) = \max\{0, -\rho\}$$
$$c_2(x) = \max\{0, -\sigma\}$$
$$c_3(x) = \max\{0, -||c - d|| + \rho + \sigma\}$$

which are zero when x is within the feasible domain.

Line search

The performance of the algorithms depends heavily on the choice of step length α_k at each iteration. The Wolfe-conditions are a popular way to perform inexact line search, and use the Armijo-condition that demands a sufficient decrease in the objective function, as well as a curvature condition that rules out unacceptable short steps.
Mathematically, the Wolfe-conditions can be formulated as follows

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k,$$
$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c2 \nabla f_k^T p_k,$$

with $0 < c1 < c2 < 1$.

A simple way to choose the step length is by backtracking line-search. Backtracking line-search ensures that the step length is short enough to satisfy the Armijo-condition, but not too short. Thus, backtracking line-search avoids the curvature condition.

Numerical experiments

Test cases

Binary classification In order to test the implementation of the algorithms, we created test cases in which data was generated and given labels according to the containing sets defined in (1) and (2).
The test cases was constructed by first generating a set of samples in $[-3, 3] \times [-3, 3] \subset \mathbb{R}^2$, and giving each sample a label $w \in \{-1, 1\}$, where $w = 1$ corresponds to the sample being inside the containing set, and $w = -1$ outside.
Further on, the samples were perturbed by some random vector with values in $[-0.5, 0.5]$ without changing the labels.

We considered two test-cases. One in which the data was labeled according to an ellipse, and another where the defining set was an hyperbola. The test cases are described by the following matrices

$$A_{\text{ellipse}} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}, \quad A_{\text{hyperbola}} = \begin{bmatrix} -1 & -3 \\ -3 & -1 \end{bmatrix},$$

and vectors b, c with element being the mean value of the samples.

From this point we will refer to the test case in which the data was labeled according to an ellipse as test case 1, while the test case defined by the hyperbola is referred to as test case 2.

Three label classification For this setting, the data points are randomly generated in the set $[-6, 6]x[-4, 4] \in \mathbb{R}^2$. The circles which label the data in the two performed tests are given in table 1.

test case	1 (overlap)	2 (no overlap)
c_{gen1}	[-3, 0]	[-3, -2]
c_{gen2}	[1, 0]	[2.5, 1]
r_1	2.5	2.5
r_2	3	3

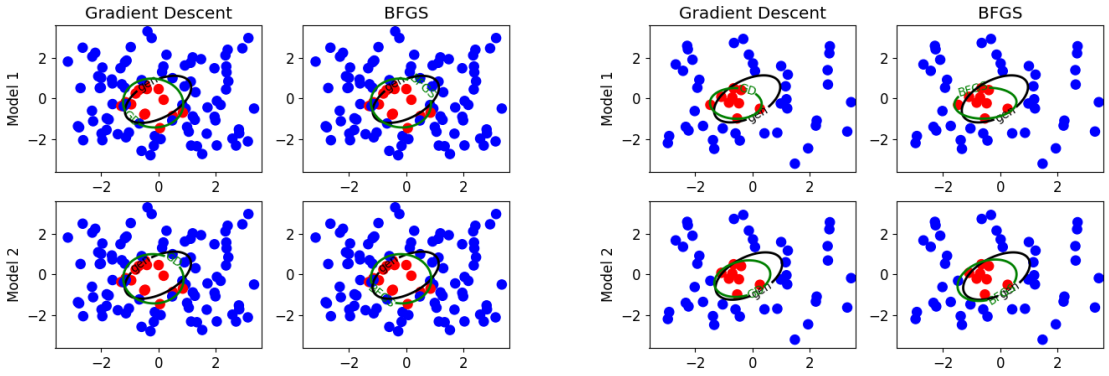
Table 1: Circles which label the data in the two test cases for the three label classification task.

Unconstrained setting

Results from test case 1 is depicted in figure 1 below. The choice of parameters in the algorithms are summarized in table 2

parameter name	value
tolerance	10^{-6}
$N_{\max}$	10 000
α_0	1.0
c_1, c_2	$10^{-4}, 0.5$
backtracking ρ	0.5
`Numpy` random seed	150

Table 2: Parameters used in the unconstrained algorithms, namely gradient descent and BFGS



(a) N = 100

(b) N = 50

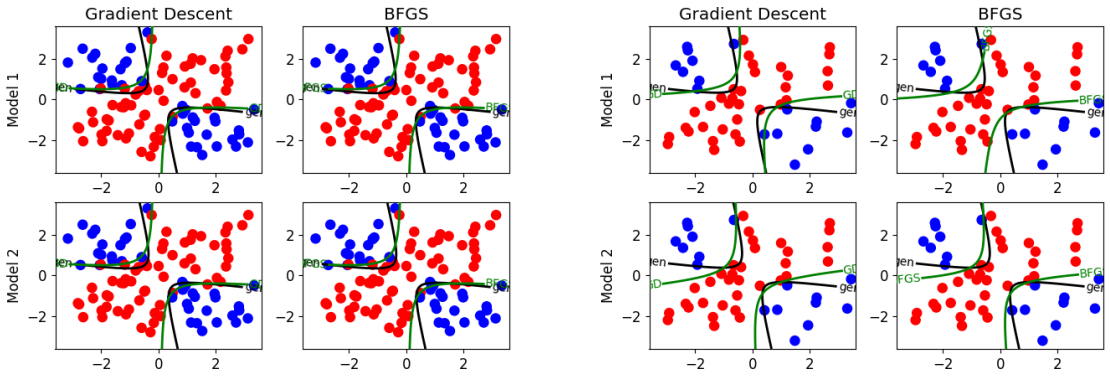
Figure 1: Minimizer of f_1 and f_2 for both algorithms with $N = 50, 100$ samples. The green ellipse depicts the solution, whereas the ellipse used to generate the data is marked in black

N = 100	Gradient Descent	BFGS
Model 1	$f_1 = 0.090$, k = 317	$f_1 = 0.090$, k= 124
Model 2	$f_2 = 0.103$, k = 428	$f_2 = 0.103$, k = 240
N = 50	Gradient Descent	BFGS
Model 1	$f_1 = 0.0$, k = 18	$f_1 = 0.0$, k = 10
Model 2	$f_2 = 0.0$, k = 2	$f_2 = 0.0$, k = 4

Table 3: Function values,f_1 and f_2 for model 1 and 2, respectively, and iterations k until termination for data generated from an ellipse and perturbed by $|\mathbf{x}| \leq \mathbf{0.5}$

Notice that the optimal solution for $N = 100$ includes misclassifications, i.e points that are not supposed to be contained in the ellipse. This is due to the rather noisy nature of the data.
For $N = 50$ the optimal ellipse separates the data completely, causing no misclassifications. This is due to the data being well-separated, even after the perturbation.
The function values and number of iterations until termination for test case 1 is summarized in table 3 above. For reference, in an ideal situation, i.e in which the labels are completely separated, the optimization results in a function value of 0.0.
Lets first consider the case with $N = 100$ samples. For model 1, observe that both algorithms converge to the same function value of 0.090. As for model 2, the algorithms converge to the same function value of 0.103. In terms of iterations, BFGS performs better than gradient descent for both models. One can also observe that the algorithms terminates to a lower function value in fewer iterations for model 1 than model 2. However, the difference is not significant.
For $N = 50$, both algorithms terminate at the optimal function value of 0.0, where we observe that model 2 terminates in fewer iterations than model 1.

Results from test case 2 is depicted in figure 2 below.



(a) N = 100 (b) N = 50

Figure 2: Minimizer of f_1 and f_2 for both algorithms with $N = 50, 100$ samples. The green hyperbola depicts the solution, whereas the hyperbola used to generate the data is marked in black

N = 100	Gradient Descent	BFGS
Model 1	$f_1 = 0.2636$, k = 811	$f_1 = 0.2636$, k = 358
Model 2	$f_2 = 0.2671$, k = 686	$f_2 = 0.2671$, k = 492
N = 50	Gradient Descent	BFGS
Model 1	$f_1 = 0.0$, k = 6	$f_1 = 0.0$, k = 23
Model 2	$f_2 = 0.0$, k = 6	$f_2 = 0.0$, k = 7

Table 4: Function values,f_1 and f_2 for model 1 and 2, respectively, and iterations k until termination for data generated from a hyperbola and perturbed by $|\mathbf{x}| \leq \mathbf{0.5}$

Observe from figure 2 that in the case of $N = 100$, the optimal solution contains misclassifications due to noisy nature of the data. However, for $N = 50$ we obtain a correct segmentation.
The function values and number of iterations until termination for test case 2 is summarized in table 4.
For $N = 100$, observe that both algorithms converge to approximately the same function value of 0.26 (within 10^{-3} accuracy) using both models. The non-zero function value is due to the misclassification of certain points. We observe that BFGS outperforms gradient descent in terms of iterations using both models. Further on, BFGS converges faster using model 1 than model 2.
As with test case 1, the algorithms using both models converge to the optimal function value of 0.0 after few iterations using $N = 50$ samples. However, we see that BFGS using model 1, the preferred situation for $N = 100$, converges slower than model 2 and gradient descent on both models.

The main difference ofthe two models is that the containing set defined by model 2 will always contain the origin, while model 1 can be translated away due to the centering vector c. This makes model 1 more flexible than model 2. One can observe this behavior in figure. 3 below.

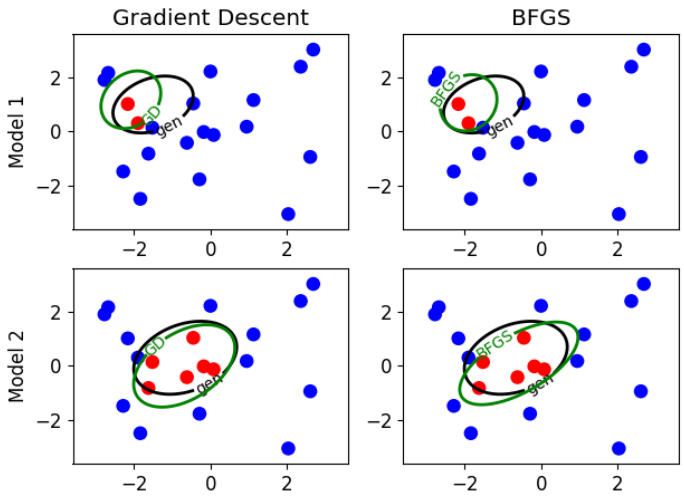


Figure 3: Minimizer of f_1 and f_2 for both algorithms with N = 20 samples. The optimal solution is marked a a green ellipse, whereas the generating set is marked in black. Observe that the optimal solution using model 2 contains the origin, while the solution for model 1 is translated away due to the centering vector c.

An advantage of model 2 is that the residuals are linear, making the gradient constant for each data point. This makes model 2 computationally simpler than model 1, in which the gradient depends on the current iterate x_k. Thus, we would expect fewer iterations when using model 2. However, we only observed this expected behavior in the case of $N = 50$.

Constrained setting

Binary classification We implemented the gradient descent algorithm with backtracking line search with the objective fucntion f_{augL} and updated μ such that $\mu_{k+1} = 100\mu_k$ and λ such that $\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$. The relevant paramters are given in table 5. The stop criteria used for gradient descent is $||x_k - x_{k-1}||_2 <$tol. Table 6 gives the results after running the algorithm with different models and datasets.

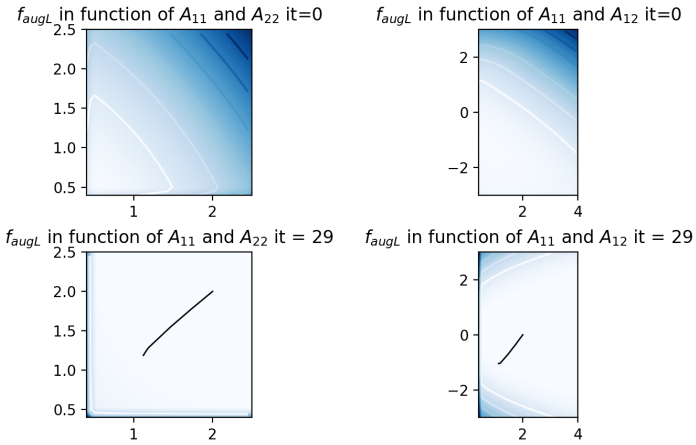


Figure 4: Level curves for f_{augL} at the start and at convergence, both in function of A_{11} and A_{22} and A_{11} and A_{12} for model 2 with parameters as given in test case 1. At convergence, the trajectory is plotted as well. The trajectory moves in south-west direction. The plot illustrates the rising boundaries in f_{augL}. In function of A_{11} and A_{22} these are two walls at $A_{11} = 0.5$ and $A_{22} = 0.5$. In function of A_{11} and A_{12} they behave according to a square root.

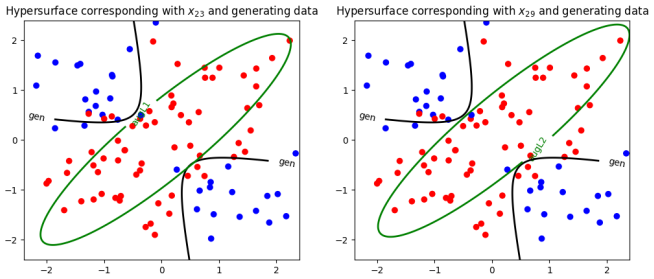


Figure 5: Hypersurface corresponding to x at convergence for model 1 (left) and model 2 (right) and data generated by a hyperbola.

The performance of the augmented Lagrangian method is indifferent between the chosen model, the number of iterations and optimal function value are close to each other. Model 1 seems be slightly advantageous in the case where the labels were generated by a hyperbola. To make sure the iterates of the augmented Lagrangian stay within the feasible region, it is important to tune the algorithm such that the added values of the constraints are high enough. With a smaller update $\mu_{k+1} = 1.1\mu_k$, the boundary did not increase fast enough and some iterates could cross the boundary. This caused some numerical instabilities because some functions where only defined in the feasible domain. Another solution to this problem would be to add a buffer to the boundary, thus making the boundaries more strict than is necessary. This algorithm is called the Bound-Constrained Lagrangian [1].

Binary classification		Three label
parameter name	value	value
tolerance	10^{-12}	10^{-15}
backtracking α_0	0.5	1.0
backtracking c	0.5	10^{-4}
backtracking ρ	0.5	0.5
nb of points	100	150
μ_0	100	10
λ_0	$[10, 10, 1]$	$[1, 1, 1]$
`Numpy` random seed	150	151

Table 5: Parameters augmented Lagrangian

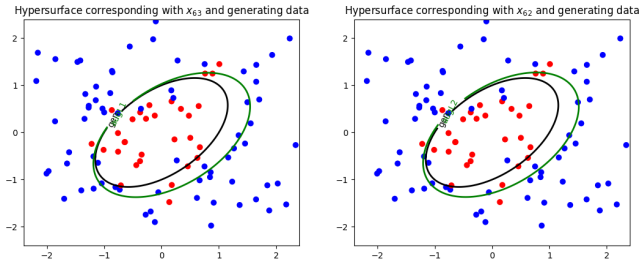


Figure 6: Hypersurface corresponding to x at convergence for model 1 (left) and model 2 (right) when the data is generated by an ellips.

	run 1	run 2	run 3	run 4
model	1	2	1	2
labels	hyperbola	hyperbola	ellips	ellips
it	23	29	63	62
fval	61	63	1.67	1.76

Table 6: Results for four runs of the augmented Lagragian algorithm with different models and labelled data. It and fval indicate the number of iterations until convergence and function value at convergence respectively.

	run 1	run 2
labels	test 1 (overlap)	test 2 (no overlap)
it	25	16
fval	80.4	2.57

Table 7: Results of the augmented lagrangian objecrtive function together with gradient descent and backtracking linesearch on the three label data segmentation problem.

Three label data segmentation We implemented the gradient descent algorithm with the augmented Lagrangian objective function for this task where we used a backtracking linesearch. The parameters are given in table 5. The results are shown in figure 7 for two test cases as described above. In the two cases the initial values are

$$c_{01} = [-3, 3], \quad c_{02} = [3, 3], \quad r_{01} = 1, \quad r_{02} = 1.$$

Based on the plot of the result, we are confident that the solution to test 1 is not the optimal solution but a local minima. Since the right circle could capture more red points if its radius were bigger without affecting the left cirlce, there exists a better situation. The constraint which forces the circles not too overlap is active here and the objective function has the shape of a steep valley. Somewhere in this valley, the backtracking linesearch algorithm returns an $\alpha \approx 0$. Fixing the step lenght and setting a fixed number of iterations resulted in a slightly better result, a function value of 20 at convergence for 60 iterations and $\alpha = 0.1$, though this algorithm is unstable since some iterates have extreme high function values.

In the case that the starting positions of the circles are switched, the circles do not manage to cross each other as is shown in figure 8.

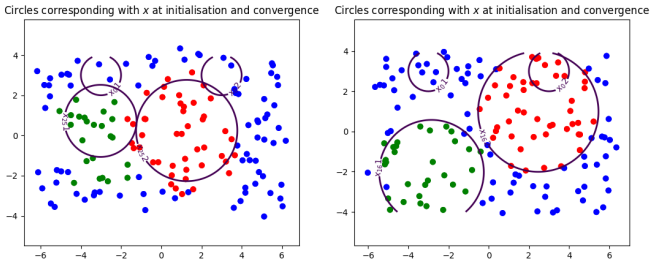


Figure 7: Circles corresponding to x_0 and x at convergence for the data separation problem.

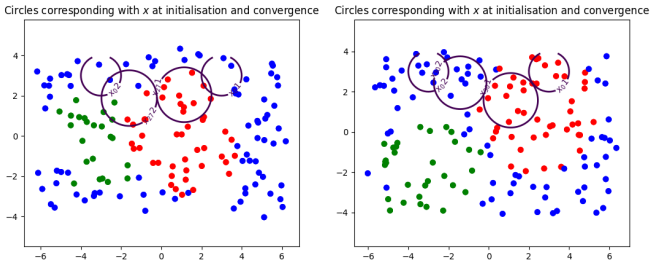


Figure 8: Circles corresponding to x_0 and x at convergence for the data separation problem. The circles started at the other side and do not manage to cross.

Summary

Both gradient descent with backtracking line-search and BFGS with bisection line-search satisfying the Wolfe-conditions performs well in the unconstrained binary classification setting. However, we only observe the expected behavior of faster convergence using model 2 in the case of $N = 50$ samples, and in test case 2 for the gradient descent method.

The augemented Lagrangian objective function in combination with gradiënt descent and backtracking line search performs well in the constrained binary classification task. Unstabilities may arise due to improper tuning

but are resolvable and by moving to buffered constraint versions avoided altogether. In the case of three label classification, the iterates get stuck in a steep valley, other, more specified algorithms would be better here.

For further studies it can be interesting to look into the scalability of the problems we have discussed. Scalability is rather important, as real-life situations contains a lot more data and dimension than we have used for our test cases. Another interesting thing to look further into is the stability of the algorithms we implemented, and how they perform on different test cases than we have presented.

References

[1] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.