Interior Point Methods: A Survey

Aditya Vikram (14040) Manuj Narang (14373) Tarun Tiwari(14751)

***Abstract*— This document reflects the evolution of interior point methods with time and how their application to solved problems revolutionized the speed for large scale data. A detailed study of interior point methods for large scale ℓ_1-regularized Least squares and Logistic regression problems is performed and their results are presented.**

I. INTRODUCTION

George Dantzig came up with the earliest algorithm: *simplex algorithm* in 1947, for solving linear programs. It starts with a vertex and proceeds by walking along an edge of the feasible polytope towards an optimum. Although this method was efficient in practice, there were no theoretical guarantees for its running time and was later proved exponential time by Klee and Minty. The first polytime algorithm: *ellipsoid method* for LP was proposed by Leonid Khachian in 1979. Despite its complexity, the ellipsoid method was much slower than the simplex method in practice.

Finally, in 1984, Karmarkar came up with a polytime method which solves about 50 times faster than the simplex method. This breakthrough marks the beginning of the interior-point revolution. [1] showed an equivalence between Karmarkars method and logarithmic barrier method for the LP in 1985. As the name suggests, interior point methods involve traversing the interior of the feasible region and the trajectory is defined by the gradient of a "barrier function" like the logarithmic barrier function. However, researchers still argued whether the two approaches were fundamentally different, or were alike, or something halfway. They began to develop other interior point methods with improved complexity bounds. Today, we know that the derivations of interior methods generally involve barrier functions' properties. Additionally, barrier methods are inherently linked to nonlinear programs, thereby allowing the extension of interior point methods to nonlinear optimization problems like quadratic programs. The evolution of interior-point methods has led to the unification of linear and nonlinear problems. Apart from the effect on convex problems, interior point revolution has paved way for fundamental advances in other fields like linear algebra, complexity theory and algorithms. Primal-dual interior point methods are the most common class of interior point methods where both primal and dual variables are updated in each step until the primal objective matches the dual.

Following is an outline of the paper: Section 2 briefly lists the related work done by other researchers; Section 3 gives a basic outline of the derivation for the primal Newton barrier method; Section 4 and 5 describe the interior point algorithms for ℓ_1-regularized least squares and logistic regression problems respectively for large-scale data along with their results. We close the paper by proposing an algorithm for a general model with ℓ_1 regularization in Section 6 followed by a Conclusion in Section 7.

II. RELATED WORK

Our work is fairly broad, so accounting for all related works is daunting. Although much work has been done in the field of interior point methods, there are very few review papers comparing different interior point algorithms. We have covered all relevant work to the best of our knowledge. In 1999, Alizadeh [2] studied the Semidefinite Programming problem and present an interior point algorithm for SDP that converges to the optimal solution in polynomial time. The authors also argue that a few existing interior point methods for LPs can be transferred to algorithms for SDP with convergence proof and time complexity bounds carrying over. Kim et.al. proposed a specialized interior point method for large scale ℓ_1-regularized least squares [3] in 2007 and Koh et.al. proposed a similar algorithm for ℓ_1-regularized least squares [4] in the same year. Our main focus will be on these two papers and their implementation. [5] presents an exhaustive literature survey of interior point methods till the year 2004 and helped us in initial phases of the project with the research direction. Our work can be somewhat thought of as an extension to [5].

III. PRIMAL NEWTON BARRIER METHOD FOR LP

First, we need to get an intuition of the connection between linear programming and barrier methods. Consider a standard LP on $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Barrier methods basically employ a penalty function for each of the constraints to convert the problem into an unconstrained optimization problem. Since the only inequality constraints here are non-negativity constraints on $\mathbf{x}$, the associated logarithmic barrier function is

$$B(\mathbf{x}, \mu) = \mathbf{c}^\top \mathbf{x} - \mu \sum_{j=1}^{n} \log(x_j)$$

where μ is called the *barrier parameter*. Using this barrier function, the LP reduces to

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} - \mu \textstyle\sum_{j=1}^{n} \log(x_j) \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b} \end{aligned}$$

Gradient and hessian of $B(\mathbf{x}, \mu)$ are given by:

$$\nabla_{\mathbf{x}} B(\mathbf{x}, \mu) = \mathbf{c} - \mu X^{-1} \mathbf{1} \qquad \nabla^2_{\mathbf{x}} B(\mathbf{x}, \mu) = \mu X^{-2}$$

where $X^{-1} = \text{Diag}(1/x_1, \cdots, 1/x_n)$. Using $\mathbf{y}$ as the lagrange vector for $A\mathbf{x} = \mathbf{b}$, we get the following:

$$\mathbf{c} - \mu X^{-1}\mathbf{1} + A^\top \mathbf{y} = \mathbf{0}$$

We use Newton's method to solve the above gradient equation. If $\delta_\mathbf{x}$ is the step length for Newton's method, the update is given by:

$$\begin{bmatrix} \mu X^{-2} & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \delta_\mathbf{x} \\ -\mathbf{y} \end{bmatrix} = \begin{bmatrix} \mu X^{-1}\mathbf{1} - \mathbf{c} \\ 0 \end{bmatrix}$$

Under the assumption that A is full rank, $AX^2A^\top$ is positive definite and we can solve the above matrix equation to get the updates for $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{x} = \mathbf{x} + \alpha\delta_\mathbf{x} = \mathbf{x} + \alpha\left(\mathbf{x} + \frac{1}{\mu}X^2\left(A^\top\mathbf{y} - \mathbf{c}\right)\right)$$
$$\mathbf{y} = \left(AX^2A^\top\right)^{-1} AX\left(X\mathbf{c} - \mu\mathbf{1}\right)$$

The sequence of these $(\mathbf{x}_\mu, \mathbf{y}_\mu)$ (subscript indicating the dependence on μ) define the *central path* or *barrier trajectory* of this LP.

IV. ℓ_1-REGULARIZED LEAST SQUARES

A. Problem Statement

The ℓ_1-regularized least squares problem involves the following optimization:

$$\min_\mathbf{x} \|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \tag{1}$$

where λ is called the regularization parameter. To gain more insight into the problem, lets have a look at the differences between ℓ_1-regularized LSP and the more common ℓ_2-regularized LSP:

1) The ℓ_1-regularized LSP yields a solution which is not linear in $\mathbf{y}$, unlike the ℓ_2-regularized LSP.
2) In ℓ_1-regularized LSP, the optimal solution $\mathbf{x}^*$ satisfies $\mathbf{x}^* = \mathbf{0}$ for $\lambda \geq \lambda_{max} = 2\left\|A^\top\mathbf{y}\right\|_\infty$ (proved in next subsection), whereas in the case of ℓ_2-regularization, convergence to zero occurs only when $\lambda \to 0$.
3) The solution to the ℓ_2-regularized LSP varies smoothly with the regularization parameter λ whereas the same is not true for ℓ_1-regularized LSP, for which the regularization path is piecewise-linear w.r.t λ.
4) ℓ_1-regularized LSP yields a sparse vector unlike the ℓ_2-regularized LSP. Sparse weight vectors are used in signal processing applications like compressed sensing.

We take a small digression to look at the formulation of ℓ_1-regularized LSP as an ℓ_1-regularized linear regression problem. Given $(\mathbf{x}_i, y_i)_{i=1}^m$ with $\mathbf{x}_i \in \mathbb{R}^n$, we want to learn a weight vector $\beta \in \mathbb{R}^n$ and bias β_0 so as to estimate y_i using $y = \beta_0 + \beta^\top\mathbf{x}$. For this estimation, we reformulate the regression problem as an ℓ_1-regularized LSP:

$$\min_{\beta,\beta_0} \left\|X^\top\beta + \beta_0\mathbf{1} - \mathbf{y}\right\|_2^2 + \lambda\|\beta\|_1$$

where $X \in \mathbb{R}^{n\times m}$ is the matrix with columns as $\mathbf{x}_i$'s and $\mathbf{y}$ is the column vector of y_i's.

B. Optimality conditions

Let us go back to the original LSP and find the optimality conditions for (1). Since the problem is unconstrained, we just need to set the gradients of the objective function to 0. But, the objective is non-differentiable at $beta_i = 0$ for any $i \in [n]$, so we use sub-gradients in place of gradients of the objective f and get:

$$\nabla_\mathbf{x} f = 2A^\top(A\mathbf{x} - \mathbf{y}) + \lambda\partial\left(\|\mathbf{x}\|_1\right) = 0$$
$$\left(2A^\top(A\mathbf{x} - \mathbf{y})\right)_i = \begin{cases} -\lambda & x_i > 0 \\ \lambda & x_i < 0 \\ [-\lambda, \lambda] & x_i = 0 \end{cases} \quad \forall i \in [n]$$

Thus, if we want to check the condition for the optimal $\mathbf{x}^* = \mathbf{0}$, putting the same in the optimality condition, we get $(2A^\top\mathbf{y})_i \in [-\lambda, \lambda] \Rightarrow \lambda \geq 2\left\|A^\top\mathbf{y}\right\|_\infty$.

C. Dual problem and Suboptimality Bound

We now derive the dual to gain more insight on the optimal solution. Since the problem is unconstrained, we introduce our constraints $\mathbf{z} = (A\mathbf{x} - \mathbf{y})$. Using ν as the lagrange dual vector for these introduced constraints, the Lagrangian is

$$\mathscr{L}(\mathbf{x}, \mathbf{z}, \nu) = \|\mathbf{z}\|_2^2 + \lambda\|\mathbf{x}\|_1 - \nu^\top(\mathbf{z} - A\mathbf{x} + \mathbf{y})$$

and the corresponding dual problem $G(\nu)$ is

$$\begin{aligned} \max_\nu \quad & -\tfrac{1}{4}\nu^\top\nu - \nu^\top\mathbf{y} \\ \text{subject to} \quad & |(A^\top\nu)_i| \leq \lambda \quad \forall i \in [n] \end{aligned}$$

The dual has an interesting property. We can define suboptimality for a given $\mathbf{x}$ by finding a dual feasible point

$$\nu = \frac{\lambda(A\mathbf{x} - \mathbf{y})}{\max_i\{(A^\top A\mathbf{x})_i - y_i\}}$$

and then calculating the duality gap w.r.this ν

$$\eta = \|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1 + \frac{1}{4}\nu^\top\nu + \nu^\top\mathbf{y}$$

Given $\mathbf{x}$, we can compute this η and say that our $\mathbf{x}$ is η-suboptimal. η can be used as one of the stopping criteria in the Newton's method discussed next.

D. The Interior Point Method

(1) can be converted to a QP by introducing new variables u_i's

$$\begin{aligned} \min_\mathbf{x} \quad & \|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\textstyle\sum_{j=1}^n u_i \\ \text{subject to} \quad & -u_i \leq x_i \leq x_i \end{aligned}$$

We define the log-barrier function for the introduced constraints in the above QP

$$\Phi(\mathbf{x}, \mathbf{u}) = -\sum_{i=1}^n \log(u_i + x_i) - \sum_{i=1}^n \log(u_i - x_i) = -\sum_{i=1}^n \log(u_i^2 - x_i^2)$$

As seen in Section 3, the central path of the interior point method comprises of the sequence of points $(\mathbf{x}(t), \mathbf{u}(t))$ that minimize the function (Using t as $1/\mu$ from Section 3):

$$\phi_t(\mathbf{x}, \mathbf{u}) = t\|A\mathbf{x} - \mathbf{y}\|_2^2 + t\lambda\sum_{j=1}^n u_i + \Phi(\mathbf{x}, \mathbf{u})$$

The Interior Point Method is as follows:

Algorithm 1: InteriorPointMethod$(\varepsilon_{\text{rel}}), \mu$

Initializations: $\mathbf{x} \leftarrow \mathbf{0}$, $\mathbf{u} \leftarrow \mathbf{1}$, $t \leftarrow \frac{1}{\lambda}$;

do

$\delta_{\mathbf{x}}, \delta_{\mathbf{u}} \leftarrow$ Newton$(\nabla^2 \phi_t(\mathbf{x},\mathbf{u})[\delta_{\mathbf{x}}, \delta_{\mathbf{u}}]^\top = -\nabla \phi_t(\mathbf{x},\mathbf{u}))$;

Let $s \leftarrow$ Step Length in previous step;

$(\mathbf{x},\mathbf{u}) \leftarrow (\mathbf{x},\mathbf{u}) + (\delta_{\mathbf{x}}, \delta_{\mathbf{u}})$;

$\eta \leftarrow$ Duality gap for a feasible dual (Section 4.C);

$$t = \begin{cases} \max\{\mu \min\{t, \frac{2n}{\eta}\}, t\} & s \geq s_{min} = 0.5 \\ t & s < 0.5 \end{cases}$$

while $\frac{\eta}{G(v)} \leq \varepsilon_{rel}$;

Note that the equation solved using Newton's method is solved approximately using Preconditioned Conjugate Gradient (PCG) steps because for large data, the exact solution takes a large amount of memory and time for the computation and might not even be feasible sometimes. Without going much into the details of PCG, it is sufficient to know that a preconditioner matrix P is used in the PCG algorithm which approximates the Hessian of $t\|A\mathbf{x}-\mathbf{y}\|_2^2 = 2tdiag(A^\top A)$, which is the computationally expensive part with diagonal entries of P. Furthermore, computing the diagonal entries can be amortized (distributed) over all interior-point iterations since we need to compute them only once. When the amortized cost of computation is still expensive, we can approximate the hessian of $diag(A^\top A)$ with a scaled identity matrix $\tau\mathbf{I}$. Typcally μ is set to be between 2 and 50, but authors make a remark that $\mu = 2$ gives good performance for most problems.

E. Results

We took a sparse signal recovery problem $\mathbf{y} = A\mathbf{x} + \varepsilon$ as our use case with a signal $x \in \mathbb{R}^{4096}$ of amplitude ± 1, generated by choosing 70 and 80 points randomly to have amplitude 1 and -1 respectively, while the rest of x is 0 (Fig. 1). The matrix A is created by first generating a matrix of size 1024×4096 generated i.i.d. from the standard normal distribution followed by orthogonalization of the rows. We tried to generate a larger dataset to test the speed of our algorithm, but we faced the hurdle of orthogonalizing such a large matrix and thus we limited our tests with 1024×4096 size. Firstly, we tried the ℓ_2-regularized least squares method and ended up with bad results because not only does it fail to predict the 1's, it also fails to get 0's because ℓ_2 is not a sparsity inducing prior. Then, we understood the implementation of our interior point algorithm from [6] and adjusted the code for our use by trying out different parameters for λ and ε_{rel} and finally choose $\lambda = 0.01$ and $\varepsilon_{\text{rel}} = 0.01$. The results are exceptional and the model predicts with very high accuracy as can be seen in Fig. 2 . We next compared the running time of the interior point algorithm with that of other matlab packages and the approximate running times are tabulated here . The runtimes are averaged over multiple runs. The results clearly show that the interior point method outperforms all other algorithms by a large margin. The predictions done by all 3 algorithms are the same.

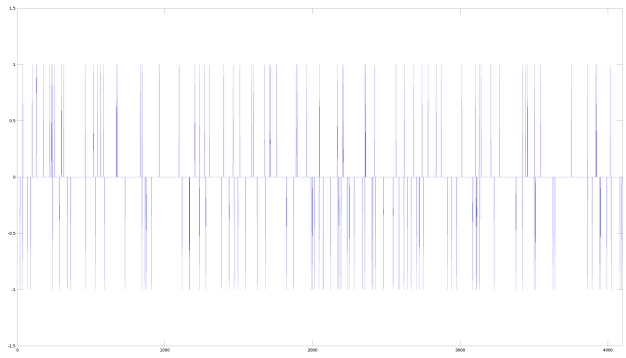

Fig. 1. Input signal

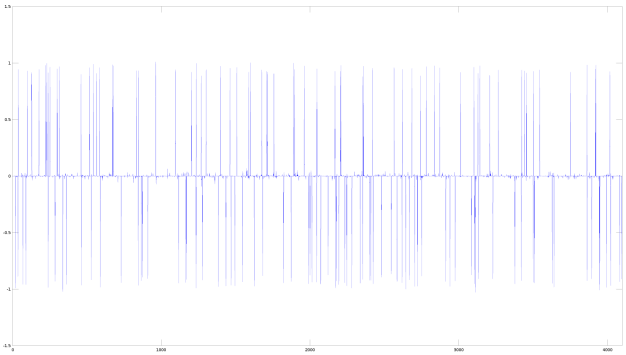

Fig. 2. Predicted signal by our esimator

Algorithm	Approximate Running time (in seconds)
Interior Point Method	10
l1-homotopy	25
l1-magic	650

V. ℓ_1-REGULARIZED LOGISTIC REGRESSION

A. Problem Statement

The ℓ_1-regularized Logistic Regression problem(LRP) is:

$$\text{minimize}_{v,w} \quad l_{avg}(v,w) + \lambda \|w\|_1$$

where, $l_{avg}(v,w)$ is the average logistic loss function given by $(1/m)\sum_{i=1}^{m} f(w^\top a_i + v b_i)$ and $f(z) = log(1 + exp(-z))$ is the logistic loss function. $\lambda > 0$ is the regularisation parameter.

It is to be noted that ℓ_1 regularised LRP is convex but not differentiable, hence solving it is computationally more challenging than ℓ_2 regularisation problem. But ℓ_1 regularisation computes a more sparse weight vector w, i.e. it has very few non-zero components. This means it selects only the relevant features to work with. This results in ℓ_1

regularisation performing better than ℓ_2 when number of features are more than number of observations.

B. Optimality conditions

We try to find the the optimality condition for the the LRP. We start by differentiating the objective function. Since the objective function is not differentiable optimality condition based on sub-differential calculus as mentioned in Bertsekas, 1999 is used.

$$\begin{aligned}\nabla_v l_{avg}(v,w) &= (1/m)\sum_{i=1}^{m} f'(w^\top a_i + vb_i)b_i \\ &= -(1/m)b^\top(1-p_{log}(v,w)) \\ \text{where,} p_{log}(v,w)_i &= exp(w^\top a_i + vb_i)/(1+exp(w^\top a_i + vb_i)) \\ & i = 1,2,..m\end{aligned}$$

$$\begin{aligned}\nabla_w l_{avg}(v,w) &= (1/m)\sum_{i=1}^{m} f'(w^\top a_i + vb_i)a_i \\ &= -(1/m)a^\top(1-p_{log}(v,w))\end{aligned}$$

The sub-differential is used for the ℓ_1 norm differentiation:

$$\begin{aligned}\partial \|w\|_i &= 1 \;\; w_i > 0 \\ &= -1 \;\; w_i < 0 \\ &= [-1,1] \;\; w_i = 0\end{aligned}$$

Necessary and sufficient condition for (v,w) pair to be optimal for LRP is:
$\nabla_v l_{avg}(v,w) = 0, \quad \nabla_w l_{avg}(v,w) + \lambda \partial \|w\|_1 = 0$

which results in:
$b^\top(1-p_{log}(v,w)) = 0$ and

$$\begin{aligned}(1/m)A^\top(1-p_{log}(v,w))_i &= \lambda \;\; w_i > 0 \\ &= -\lambda \;\; w_i < 0 \\ &= [-\lambda,\lambda] \;\; w_i = 0\end{aligned}$$

If we consider optimal solution for w=0, i.e. pair of the form (v,0), the first condition above gives $v = log(m_+/m_-)$, where m_+ and m_- is number of training example which has outcome 1 and 0 respectively. From second condition,we have
$\left\|(1/m)A^\top(1-p_{log}(v,0))\right\|_{\text{inf}} \leq \lambda$

From this we get an upper bound on regularisation parameter λ that if lambda is greater than above value all the elements of w will be zero i.e. we get maximally sparse w vector.

C. Dual

To write the dual, we first introduce a new variable z and new equality constraint $z_i = w^\top a_i + vb_i$ for i = 1,2,..m. The problem becomes:

$$\begin{aligned}\text{minimize } & (1/m)\textstyle\sum_{i=1}^{m} f(z_i) + \lambda \|w\|_1 \\ \text{subject to } & z_i = w^\top a_i + vb_i\end{aligned}$$

Let θ_i be the dual variable corresponding to the equality z_i. The langrangian for the problem can be written as:

$$L(v,w,z,\theta) = (1/m)\textstyle\sum_{i=1}^{m} f(z_i) + \lambda \|w\|_1 + \theta^\top(-z - Aw - bv)$$

The dual function $\inf_{\mathbf{v},w,z} L(v,w,z,\theta)$ is given by:

$$\begin{aligned}G(\theta) &= -(1/m)\sum_{i=1}^{m} f^*(-m\theta_i) \;\; \left\|A^\top\theta\right\|_\infty \leq \lambda, b^\top\theta = 0 \\ &= -\infty \; otherwise\end{aligned}$$

where $f^*()$ is the conjugate of the logistic function. We have now the following dual function:

$$\begin{aligned}&\text{maximize } G(\theta) \\ &\text{subject to } \left\|A^\top\theta\right\|_\infty \leq \lambda, b^\top\theta = 0\end{aligned}$$

Since the LRP follows slater's condition there exist an optimal dual θ^*, which satisfies $G(\theta^*) = p^*$
Also we can relate the dual optimal with primal optimal point (v^*, w^*) by:

$$\theta^* = (1/m)(1-p_{log}(v^*,w^*))$$

D. Interior point method

Since we have a non-differentiable objective function, to apply interior point we will transform objective as follows:

$$\begin{aligned}&\text{minimize } l_{avg}(v,w) + \lambda 1^\top u \\ &\text{subject to } -u_i \leq w_i \leq u_i, i = 1,2,...n\end{aligned}$$

We will define the logarithmic barrier function :

$$\begin{aligned}\phi(w,u) &= -\sum_{i=1}^{n} log(u_i + w_i) - \sum_{i=1}^{n} log(u_i - w_i) \\ &= -\sum_{i=1}^{n} log(u_i^2 - w_i^2)\end{aligned}$$

We augment the objective function with barrrier function and remove the constraints. New objective function becomes

$$F_t(v,w,u) = t l_{avg}(v,w) + t\lambda 1^\top u + \phi(w,u)$$

where $t > 0$ is a parameter. The function will have a unique minimum as it is strictly convex and bounded below. Let the minimum point be $(v^*(t), w^*(t), u^*(t))$. For different t we have different minimum point, the path followed by those point is the central path. For each $(v^*(t), w^*(t), u^*(t))$ we will associate $\theta^*(t) = (1/m)(1-p_{log}(v^*(t), w^*(t)))$, which can be shown to be dual feasible.

Once we have $\theta*$ we can find $G(\theta^*)$ and hence the duality gap:

$$l_{avg}(v^*(t), w^*(t)) + \lambda \|w^*(t)\|_1 - G(\theta^*)$$

Also we know that the $(v^*(t), w^*(t))$ cannot be more than $2n/t$ sub-optimal. Hence we have a constraint :

$$l_{avg}(v^*(t), w^*(t)) + \lambda \|w^*(t)\|_1 - G(\theta^*) \leq l_{avg}(v^*(t), w^*(t)) + \lambda 1^\top u^*(t) - G(\theta^*) = 2n/t$$

Now to calculate the optimum point we use the following algorithm:

1) For a given t we calculate $(v^*(t), w^*(t), u^*(t))$ using Newton's method to minimize $F_t(v, w, u)$ starting with random initial point. Let s be step length for this Newton's step.
2) Evaluate the duality gap η by finding a dual feasible point.
3) Next we update t as follows:

$$t = \begin{cases} \max\{\mu \min\{t, \frac{2n}{\eta}\}, t\} & s \geq s_{min} = 0.5 \\ t & s < 0.5 \end{cases}$$

   Again repeat step one, with optimum point calculated before as the new initial point.
4) Repeat the steps until convergence.

A termination criteria can be $2n/t \leq \varepsilon$, ε being the a chosen tolerance.

VI. A General Algorithm

When faced with a smooth differentiable objective function along with ℓ_1-regularization, we propose an interior point algorithm by observing the general pattern of these two algorithms we've seen. We need not worry about updating the parameter t as the authors have claimed that the default settings work for a wide range of problems. We need to make changes in the equation to solve using Newton's method. Suppose the optimization objective is

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \lambda \|\mathbf{x}\|_1$$

Then, after incorporating the log-barrier function by introducing new constraints, we get the central path by minimizing

$$\phi_t(\mathbf{x}, \mathbf{u}) = \mathrm{t} f(\mathbf{x}, \mathbf{y}) + t\lambda \sum_{j=1}^{n} u_i - \sum_{i=1}^{n} \log(u_i^2 - x_i^2)$$

For using Newton's method, we need the gradient and Hessian of $\phi_t(\mathbf{x}, \mathbf{u})$.

$$\nabla_{\mathbf{x}} \phi_t(\mathbf{x}, \mathbf{u}) = \mathrm{t} \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + 2U^{-1}\mathbf{x} \qquad (U = diag(u_i^2 - x_i^2))$$

$$\nabla_{\mathbf{u}} \phi_t(\mathbf{x}, \mathbf{u}) = \mathrm{t} \nabla_{\mathbf{u}} f(\mathbf{x}, \mathbf{y}) + t\lambda \mathbf{1} - 2U^{-1}\mathbf{u}$$

The hessian can also be computed similarly and used for Newton's method given that the f is smooth and the gradients are computable with standard tools. Rest of the algorithm has similar outline as Algorithm 1 .

VII. Conclusion

We started out by discussing the evolution of interior point methods and the contributions by researchers like Karmarkar. Then, we saw how to apply the primal newton's barrier method for LP and its close connections with LP. We also saw the theory behind the interior point methods for ℓ_1-regularized least squares problem and ℓ_1-regularized logistic regression problem and how interior point methods can be adjusted for large-scale data by approximating the Newton's step and still getting reliable results. In addition, we implemented the ℓ_1-regularized least squares formulation for a use case: *compressed sensing* with a toy dataset and verified the results of [3]. The interior point method which we discussed outperforms other algorithms like l1-magic by a large margin in terms of running time. Lastly, we concluded by proposing our algorithm for a general regression problem with ℓ_1 regularization, which should work for a wide variety of smooth loss functions. In future, we would like to verify the same on some popular loss functions used in convex optimization and machine learning.

References

[1] P. E. Gill et.al. "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method". In: *Math. Prog. 36* (1986).

[2] F. Alizadeh. "Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization". In: *SIAM Journal on Optimization* (1995). DOI: 10.1137/0805002.

[3] K. Koh S.J. Kim S. Boyd. "An Interior-Point Method for Large-Scale ℓ_1-Regularized Least Squares". In: *IEEE Journal on Selected Topics in Signal Processing* (2007).

[4] Seung-Jean Kim Kwangmoo Koh and Stephen Boyd. "An Interior-Point Method for Large-Scale ℓ_1-Regularized Logistic Regression". In: *Journal of Machine Learning Research* (2007).

[5] M. H. Wright. "The interior-point revolution in optimization: History, recent developments, and lasting consequences". In: *The Bulletin of the American Mathematical Society* (2005).

[6] *l1_ls: Simple Matlab Solver for l1-regularized Least Squares Problems*. URL: https://web.stanford.edu/~boyd/l1_ls/.