CPSC536M: Assignment 2

Vincent Guan: 18533281
Joseph Janssen: 28738152
Gabriel Jarry-Bolduc: 93424166

1 Introduction

Multiple regression is often diminished by issues such as high variance, models that are too large to interpret, and models that overfit data noise (Tibshirani, 1996). Several additions have been made to multiple regression to fix these issues including the non-negative garotte, the lasso, elastic net penalization, and stability selection (Breiman, 1993; Tibshirani, 1996; Zou & Hastie, 2005; Meinshausen & Buhlmann, 2010). In this short paper, we focus on the lasso. There are several methods referenced in Tibshirani, (1996) to solve the lasso including those suggested by Lawson & Hansen, (1974) and a slightly more efficient one by David Gay; however both of these methods are sensitive to a large number of predictor variables. The Frank-Wolfe algorithm (also known as the conditional gradient method) can also be used to solve the lasso problem (Jaggi, 2013; Frank & Wolfe, 1956). The Frank-Wolfe algorithm has the advantages that it scales well, provides low-rank and sparse solutions, and is guaranteed to converge in $O(\frac{1}{\epsilon})$ iterations for some tolerance level ϵ.

Remote sensing has been a huge topic of interest in many scientific disciplines such as hydrogeology with the use of satellite imagery and artificial intelligence with the use of self-driving cars. The raw image that is remotely sensed, however, can include unwanted artifacts from atmospheric disturbances, blurring, or defocusing that must be removed before the data is usable (Lim et al., 2020). The lasso can be used to resolve this issue. The goals of this paper are three-fold. First, we summarize the paper titled *Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization* by Martin Jaggi. Then, the Frank-Wolfe algorithm is described and it is applied to an image deblurring problem inspired from Section 5.1 in (Beck & Teboulle, 2009). Last, we test the sensitivity of our results to regularization magnitude.

2 Brief summary

We begin by discussing the main results of Jaggi, (2013), which considers the optimization problem

$$\min_{x \in C} f(x). \tag{1}$$

C is a nonempty convex, compact subset of a vector space, and f is a convex function satisfying $C \subseteq \operatorname{dom} f$. Also, we assume that $\operatorname{dom} f$ is open and f is continuously differentiable on $\operatorname{dom} f$. Note that the continuity of f ensures that f does in fact attain its minimum on the compact subset C. One of the first iterative methods developed to solve this type of problem is the Frank-Wolfe method (Algorithm 1).

Algorithm 1: General Frank-Wolfe

Result: $x^{(K)}$
Choose $x^{(0)}$ such that $x^{(0)} \in C$;
for $k = 0...K$ **do**
  $s^{(k)} \in \operatorname{argmin}_{y \in C} \left\langle y, \nabla f(x^{(k)}) \right\rangle$;
  $x^{(k+1)} = (1 - \frac{2}{k+2})x^{(k)} + \frac{2}{k+2} s^{(k)}$;
end

The main achievement of the paper is to present stronger convergence results for Frank-Wolfe type algorithms for convex constrained optimization that hold in a more general setting. This is achieved by defining the duality gap equation

$$g(x) = \max_{s \in C} \langle x - s, \nabla f(x) \rangle \tag{2}$$

where x is a feasible point. Equation (2) can be viewed as a measure of the quality for the incumbent solution $x^{(k)}$. Indeed, for any $k \in \mathbb{N}$, we have $f(x^{(k)}) - f(x^\star) \leq g(x^{(k)})$ where $x^\star$ is the true minimizer. In general, the minimizer $x^\star$ is unknown; however, the duality gap is usually easy to compute. Jaggi introduces three additional variants of the Frank-Wolfe algorithm. In the first version, the subproblem $\operatorname{argmin}_{s \in C} \left\langle s, \nabla f(x^{(k)}) \right\rangle$ is approximately solved. This approach is valuable when the exact linear subproblem is too expensive to solve. Instead of using the same value of $\gamma = \frac{2}{t+2}$ at each iteration, the second variant uses line search at each iteration to select γ. The third variant gives more flexibility on the choice for $x^{(k+1)}$. The value of $x^{(k+1)}$ is found by minimizing the objective function f over the convex

hull of all search directions $s^{(k)}$ used so far. The two last variants are developed in the hope of making more progress per iteration. The downside is that each iteration can become harder to solve than the original problem. One of the main results of the paper (Theorem 2) is to show that these three variants have bounded duality gaps. Moreover, Jaggi realizes that the same convergence results can be obtained by using an approximation of the gradient instead of the true gradient. He also demonstrates that the convergence analysis is fully invariant under affine transformations and reparametrizations of the domain C. Lastly, the article analyzes the Frank-Wolfe algorithm when C is an atomic set and the wide range of applications are demonstrated. One advantage of the Frank-Wolfe algorithm is that each iteration uses only one atom, so sparse solutions are returned. In that sense, the inexpensive linear subproblem involved in Algorithm 1 can be viewed favorably compared to the quadratic subproblems involved in more sophisticated methods such as *projected gradient* (Beck, 2017, Section 8.2) and *proximal methods* (Beck, 2017, Chapter 10). In fact, the complexity of optimizing over some of the most common domains including all ℓ_p-norm balls and sparse non-negative vectors only require linear time complexity for each iteration.

3 The Frank-Wolfe Method

Before testing the Frank-Wolfe method on the deblurring example, let us provide more information about the method. To illustrate the mechanism of Algorithm 1, we compare it to the projected gradient method. The update rule for the projected gradient method is

$$x^{(k+1)} = \mathrm{P}_C(x^{(k)} - \gamma^{(k)} \nabla f(x^{(k)}))$$

where P_C denotes the orthogonal projection operator onto the set C and $\gamma^{(k)}$ is the step size at iteration k. Hence, this method requires the computation of a projection at each iteration. On the other hand, the Frank-Wolfe method update rule is a convex combination of the incumbent solution $x^{(k)}$ and a minimizer of a linearization of the function f over C. Thus, we conclude that if the computation of a minimizer of a linear function over C is an easier task to achieve than computing the orthogonal projection onto C, then the Frank-Wolfe algorithm should be chosen. Finding the minimizer of a linearization of the function f is certainly a simple task in the following example.

Let us consider the image deblurring problem with an $m \times n$ image, which may be represented as an $mn \times 1$ vector, x. The image vector x can be blurred by multiplying it with an $mn \times mn$ blurring matrix A, i.e. $b = Ax$, where b is the blurred image. To solve this, we would like to take A^{-1}, but we may not assume that A is well-conditioned. Instead, we can solve for x by minimizing the constrained least squares problem with the regularization parameter $\tau \in \mathbb{R}^+$:

$$\min_x \frac{1}{2}||Ax - b||_2^2 \ s.t. \ ||x||_1 \leq \tau. \tag{3}$$

Note that (3) can be written as

$$\min_{x \in \tau \, \mathrm{B}_1} \frac{1}{2}||Ax - b||_2^2 \tag{4}$$

where B_1 is the unit ℓ_1-norm ball. Let $g^{(k)} = \nabla f(x^{(k)})$ where $\nabla f(x) = A^T(Ax - b)$. The subproblem is $s \in \operatorname{argmin}_{y \in \tau \, \mathrm{B}_1} \langle y, g^{(k)} \rangle$. Let j be the smallest index in $\{1, 2, \ldots, n\}$ such that $[g^{(k)}]_j = \|g^{(k)}\|_\infty$. Then the solution of the subproblem is given by

$$s^{(k)} = -\mathrm{sign}([g^{(k)}]_j) \tau e_j. \tag{5}$$

Choosing the smallest index j provides a simple rule to choose a unique solution to the subproblem since more than one may exist. Algorithm 2 presents the Frank-Wolfe method applied to problem of the form (3).

Algorithm 2: Frank-Wolfe for image deblurring

Result: $x^{(K)}$
Choose $x^{(0)}$ such that $||x^{(0)}||_1 \leq \tau$;
for $k = 0 \ldots K$ **do**
  $s^{(k)} \in \operatorname{argmin}_{y \in \tau \, \mathrm{B}_1} \left\langle y, A^T(Ax^{(k)} - b) \right\rangle$;
  $x^{(k+1)} = \left(1 - \frac{2}{k+2}\right) x^{(k)} + \frac{2}{k+2} s^{(k)}$;
end

4 Numerical Experiment

In this section, we use the image *mri.tif* from the Image Processing Toolbox in MATLAB. This is a grayscale image of size 128×128. The matrix associated with the image is scaled so that the greatest entry is equal to 1. A Gaussian blur of size 9×9 and standard deviation 4 is applied to the original image. The original image and the blurred image are illustrated in the Appendix (Figure 1). We consider the problem in (4) where A represents the blur operator and b is the blurred image (vectorized). The process to obtain A is well explained by Hansen in Hansen, Chapter 4 and due to page restrictions, we will not elaborate on this subject in this present report (see Matlab code for more details). The coding done in Matlab to obtain the matrix A was inspired by the coding by Hansen available at HNO. For the algorithm to run faster, the blurred image B is kept as a matrix in $\mathbb{R}^{128\times128}$. The matrix A and X are also in $\mathbb{R}^{128\times128}$. The starting point $X^{(0)}$ used on every problem is the zero matrix in $\mathbb{R}^{128\times128}$. The experiment is repeated 10 times for different values of τ. It is also repeated for different number of iterations: $K = 100, 200, 1000, 10000$. The following table presents the function value obtained at iteration K for each case.

Table 1: Frank-Wolfe algorithm applied to an image deblurring problem

τ	$K = 100$	$K = 200$	$K = 1000$	$K = 20000$
0.1	1.26e03	1.26e03	1.26e03	1.26e03
1	1.21e03	1.21e03	1.21e03	1.21e03
10	8.31e02	8.31e02	8.31e02	8.31e02
50	3.58e01	3.49e01	3.43e01	3.43e01
70	2.75e00	6.42e-01	3.59e-01	9.30e-04
75	3.02e00	6.17e-01	3.91e-02	1.50e-04
80	3.21e00	9.16e-01	2.46e-02	1.40e-04
100	4.43e00	1.10e00	4.65e-02	1.70e-04
200	2.36e01	5.58e00	1.74e-01	5.10e-04
1000	8.94e02	1.80e02	8.52e00	1.11e-02

Note that the best results in each column are indicated in blue. Figure 2 in the Appendix illustrates the evolution of the image at each K when $\tau = 75$. We see that when $K = 20000$ and $\tau = 75$, the image obtained is similar to the original image. Based on our results, it seems that the best value of τ is in the interval $[75, 100]$ if we run the algorithm for 20000 iterations. Note that the norm of the original image is approximately 51. Figure 3 in the Appendix illustrates that when τ is too small, the minimizer found by the algorithm is almost the zero matrix and when τ is too large, the result looks similar to the result of $\tau = 75$ with only 1,000 iterations. Indeed, the choice of τ is important.

5 Final Remarks

We showed that the Frank-Wolfe algorithm can be applied to solve an image deblurring problem of the form (4). Based on our numerical experiment, the value of τ has a significant impact on the quality of the solution. To obtain a solution that is similar to the original image, the Frank-Wolfe Algorithm needed 20000 iterations. This simple experiment showed the limitation of the algorithm. Indeed, the image considered is extremely small and does not contain colour. Moreover, if we compare our result to the FISTA algorithm in (Beck, A. & Teboulle, M., 2009), we see that Frank-Wolfe does poorly in terms of number of iterations. The number of iterations necessary to obtain a "good" solution with Frank-Wolfe could be decreased by considering a better starting point than the zero matrix or by using line search to choose the step size at each iteration. The zero matrix was considered since it provides a feasible starting point for any value of τ. Future directions to explore include conducting the same experiment with color images and comparing Frank-Wolfe with the projected gradient algorithm or other more advanced algorithms. Efficient strategies for finding the ideal τ given a fixed number of iterations could also be investigated.

6 References

Beck, A. (2017). First-order methods in optimization. Society for Industrial and Applied Mathematics.

Beck, A., & Teboulle, M. (2009). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE transactions on image processing, 18(11), 2419-2434.

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th international conference on machine learning (No. CONF, pp. 427-435).

Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. Naval research logistics quarterly, 3(1-2), 95-110.

Osborne, M. R., Presnell, B., & Turlach, B. A. (2000). A new approach to variable selection in least squares problems. IMA journal of numerical analysis, 20(3), 389-403.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Breiman, L. (1993). Better subset selection using the non-negative garotte. Technical Report. University of California, Berkeley.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2), 301-320.

Meinshausen, N., & Bühlmann, P. (2010). Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), 417-473.

Lim, H., Yu, S., Park, K., Seo, D., & Paik, J. (2020). Texture-Aware Deblurring for Remote Sensing Images Using ℓ_0-Based Deblurring and ℓ_2-Based Fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 3094-3108.

Beck, A., Taboulle, M.(2009) A fast shrinkage-thresholding algorithm for linear inverse problems. SiAM Journal on imaging sciences, 183-202.

Hansen, P. C., Nagy, J. G., O'leary, D. P. (2006). Deblurring images: matrices, spectra, and filtering. Society for Industrial and Applied Mathematics.

Appendix

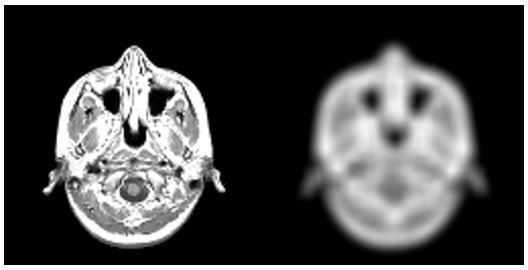

Figure 1: The original image and the blurred image

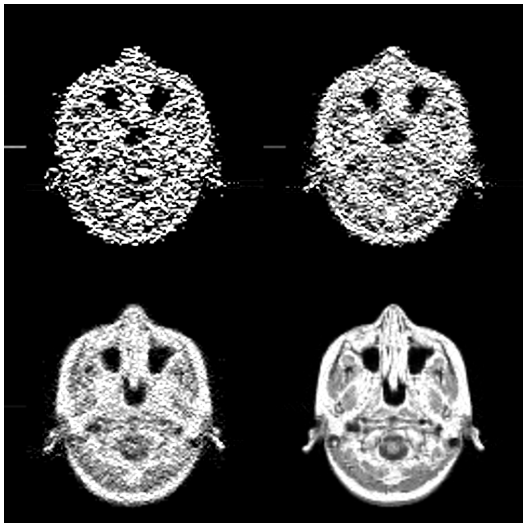

Figure 2: The evolution of the image when $\tau = 75$ for $K = 100, 200, 1000, 20000$ iterations.

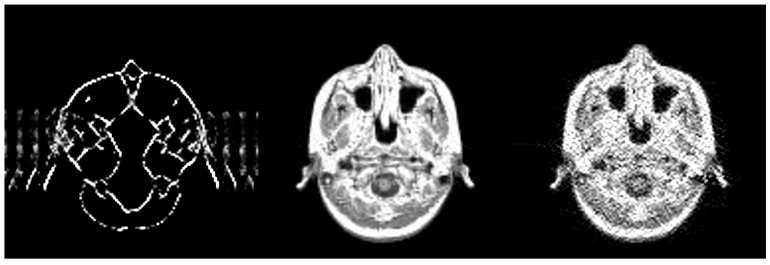

Figure 3: The deblurred image result after 20000 iterations when $\tau = 10, 100, 1000$.