6 Supplementary Material

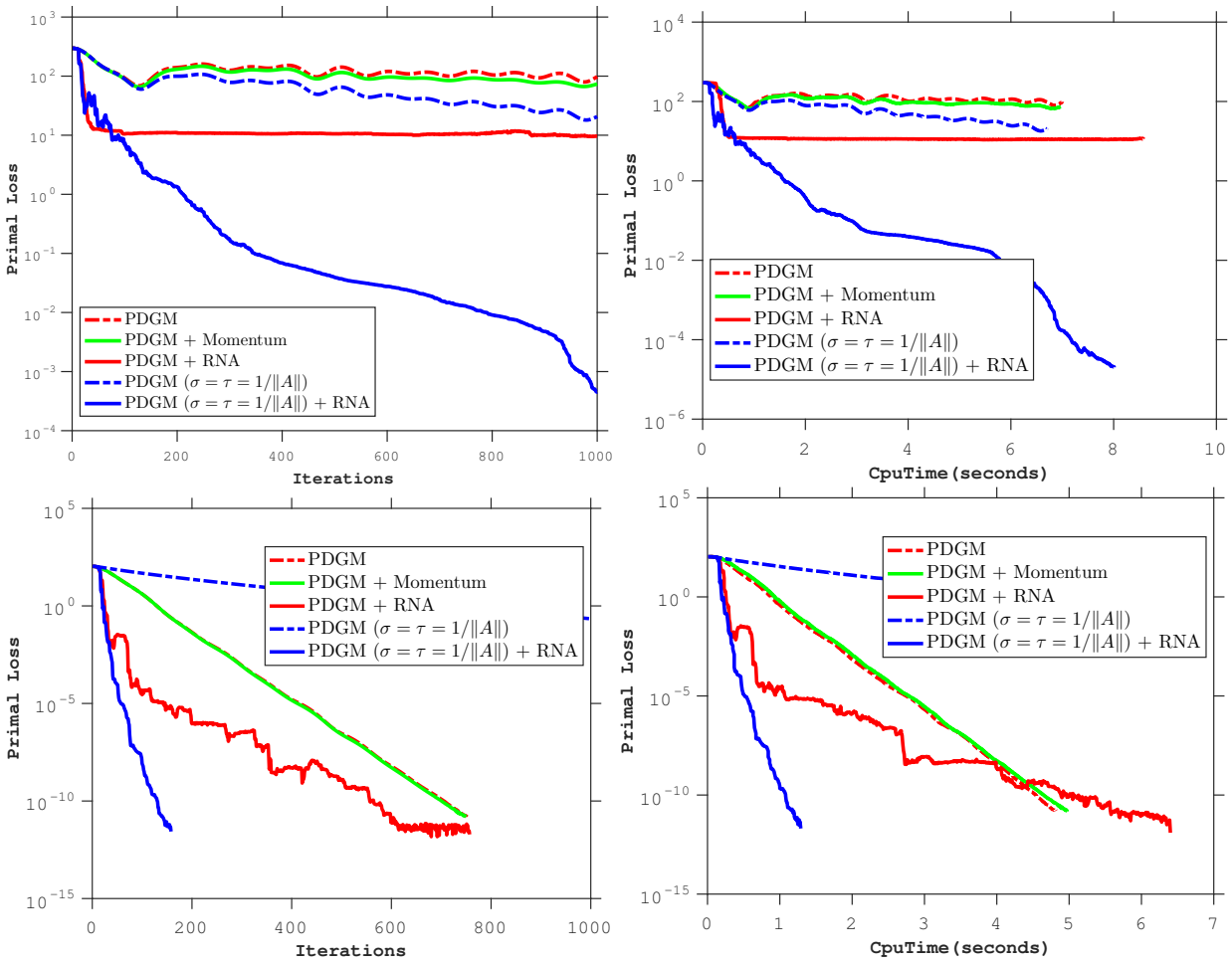

Figure 4: Quadratic loss on the Madelon dataset (Guyon, 2003). Top: $\mu = 10^{-2}$. Bottom: $\mu = 10^{2}$. Left: Iterations. Right: Time. Comparison of online RNA with other variants of primal-dual gradient methods.

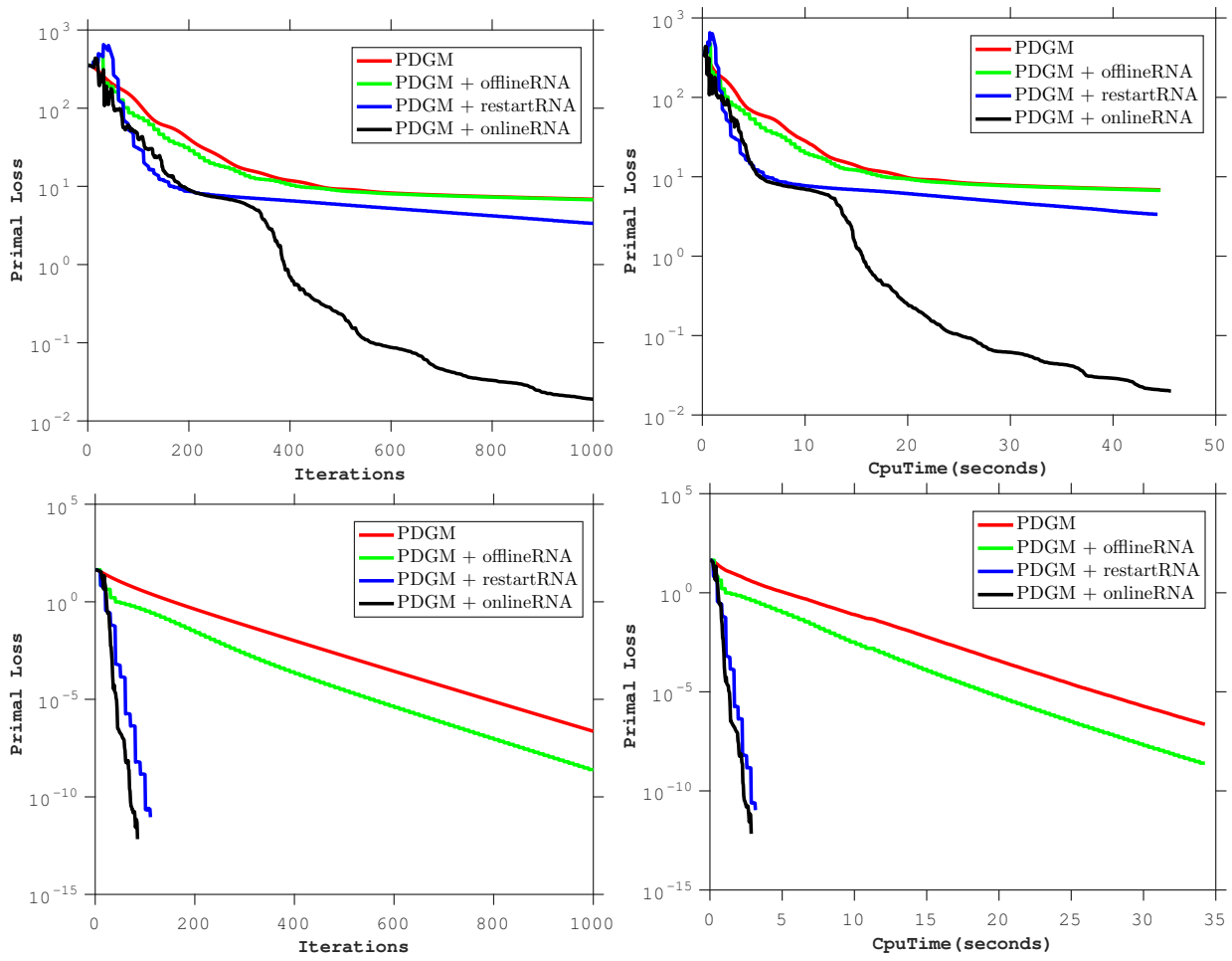

Figure 5: Logistic loss on the Madelon dataset (Guyon, 2003). Top: $\mu = 10^{-2}$. Bottom: $\mu = 10^{2}$. Left: Iterations. Right: Time. Comparison of offline, restart and online variants of RNA on primal-dual gradient methods.

6.1 Theorems and Proofs

Theorem 6.1 *(Johnson, 1974) For any real 2 by 2 matrix*

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the boundary of the numerical range is an ellipse whose axes are the line segments joining the points x to y and w to z respectively where,

$$x = \frac{1}{2}(a + d - ((a-d)^2 + (b+c)^2)^{1/2})$$
$$w = \frac{a+d}{2} - i\left|\frac{b-c}{2}\right|$$
$$y = \frac{1}{2}(a + d + ((a-d)^2 + (b+c)^2)^{1/2})$$
$$z = \frac{a+d}{2} + i\left|\frac{b-c}{2}\right|$$

are the points in the complex plane.

Theorem 6.2 *(Fischer and Freund, 1991, Th. 2) Let $k \geq 5$, $r > 1$ and $c \in \mathbb{R}$. The polynomial*

$$T_{k,\kappa}(z) = T_k(z)/T_k(1-\kappa)$$

is the unique solution of problem (12) if either

$$|1-\kappa| \geq \frac{1}{2}\left(r^{\sqrt{2}} + r^{-\sqrt{2}}\right)$$

or

$$|1-\kappa| \geq \frac{1}{2a_r}\left(2a_r^2 - 1 + \sqrt{2a_r^4 - a_r^2 + 1}\right)$$

where $a_r = (r + 1/r)/2$.

Theorem 6.3 *The numerical range of operator O is given as the convex hull of the numerical ranges of the operators O_j, i.e.*

$$W(O) = \mathbf{Co}\{W(O_1), W(O_2), \cdots, W(O_n)\} \tag{24}$$

Proof. Let $v_1, v_2, \cdots, v_n$ be eigen vectors associated with eigen values $\lambda_1, \lambda_2, \cdots, \lambda_n$ of the matrix A. We can write

$$A = \sum_{j=0}^{n} \lambda_j v_j v_j^T \qquad I = \sum_{j=0}^{n} v_j v_j^T$$

Let $t \in W(O) \subset \mathbb{C}$. By definition of the numerical range, there exists $z \in \mathbb{C}^{2n}$ with $z^* z = 1$ and

$$\begin{aligned}
t &= z^* \begin{bmatrix} 0 & A \\ -\beta I & (1+\beta)A \end{bmatrix} z \\
&= z^* \begin{bmatrix} 0 & \sum_{j=1}^n \lambda_j v_j v_j^T \\ -\beta \sum_{j=1}^n v_j v_j^T & (1+\beta)\sum_{j=1}^n \lambda_j v_j v_j^T \end{bmatrix} z \\
&= \sum_{j=0}^{n} z^* \left(\begin{bmatrix} 0 & \lambda_j \\ -\beta & (1+\beta)\lambda_j \end{bmatrix} \otimes v_j v_j^T \right) \mathbf{vec}([z_1, z_2]) \\
&= \sum_{j=0}^{n} z^* \, \mathbf{vec}\left(v_j v_j^T [z_1, z_2] \begin{bmatrix} 0 & \lambda_j \\ -\beta & (1+\beta)\lambda_j \end{bmatrix}^T \right)
\end{aligned}$$

and since $v_j v_j^T v_j v_j^T = v_j v_j^T$, this last term can be written

$$\begin{aligned}
t &= \sum_{j=0}^{n} \mathbf{Tr}\left(v_j v_j^T [z_1, z_2] \begin{bmatrix} 0 & \lambda_j \\ -\beta & (1+\beta)\lambda_j \end{bmatrix}^T [z_1, z_2]^* v_j v_j^T \right) \\
&= \sum_{j=0}^{n} \mathbf{Tr}(v_j v_j^T) \left([v_j^T z_1, v_j^T z_2] \begin{bmatrix} 0 & \lambda_j \\ -\beta & (1+\beta)\lambda_j \end{bmatrix}^T [z_1^* v_j, z_2^* v_j]^T \right)
\end{aligned}$$

Now, let $w_j = [z_1^* v_j, z_2^* v_j]^T$, and

$$y_j = \frac{w_j^T O_j w_j}{\|w_j\|_2^2}$$

and by the definition of the numerical range, we have $y_j \in W(O_j)$. Therefore,

$$t = \sum_{j=0}^{n} \left(\frac{w_j^T O_j w_j}{\|w_j\|_2^2} \right) \|w_j\|_2^2$$

hence

$$t \in \mathbf{Co}(W(O_1), W(O_2), \cdots, W(O_n)).$$

We have shown that if $t \in W(O)$ then $t \in \mathbf{Co}(W(O_1), W(O_2), \cdots, W(O_n))$. We can show the converse by following the above steps backwards. That is, if $t \in \mathbf{Co}(W(O_1), W(O_2), \cdots, W(O_n))$ then we have,

$$t = \sum_{j=0}^{n} \theta_j \left(\frac{w_j^T O_j w_j}{\|w_j\|_2^2} \right)$$

where $\theta_j > 0$, $\sum_{j=0}^{n} \theta_j = 1$ and $w_j \in \mathbb{C}^2$. Now, let

$$z = \sum_{j=0}^{n} \frac{\mathbf{vec}(v_j w_j^T)\theta_j^{1/2}}{\|w_j\|}$$

and we have,

$$t = \sum_{j=0}^{n} [z_1^* v_j z_2^* v_j] O_j \begin{bmatrix} v_j^T z_1 \\ v_j^T z_2 \end{bmatrix}$$

wherein we used the fact that $v_j^T v_k = 0$ for any $j \neq k$ and $v_j^T v_j = 1$ in computing $w_j^T = [z_1^* v_j z_2^* v_j]$. We also note that $z^* z = 1$ by the definition of z and rewriting the sum in the matrix form we can show that $t \in W(O)$ which completes the proof. ■

6.2 Online RNA

In this section, we develop the online-RNA algorithm, which injects the extrapolated point built by RNA directly inside the algorithm. This means accelerating and restarting the algorithm at each step, thus improving its rate of convergence.

This modification of RNA procedure was introduced in (Scieur et al., 2018), and consist of coupling the algorithm $g(x)$ with the RNA step as follow,

$$\begin{cases} x_{k+1} &= g(y_k), \\ y_{k+1} &= \sum_{i=1}^{k+1} c_i x_i, \end{cases} \tag{25}$$

where coefficients c_i are computed using RNA algorithm 1 with residues $x_{k+1} - y_k$ instead of $x_{k+1} - x_k$.

6.3 Extension to nonlinear functions

It is possible to extend our results on quadratic functions to nonlinear functions, as done in (Scieur et al., 2016) and (Scieur et al., 2017). Convergence of RNA in the general case is essentially obtained via a perturbation analysis of the quadratic case. Indeed, at a step k, consider the following perturbed linear iteration

$$g(x_k) = A(x_k - x^*) + x^* + \epsilon_k \tag{26}$$

where ϵ_k corresponds to any kind of perturbation (e.g. non-linear or stochastic residual noise), and A corresponds to the Hessian of the operator $g(x)$ at the fixed-point $x^* = g(x^*)$. This perturbed linear iteration corresponds to a perturbation of (3), and includes for example, SGD+momentum steps on a non-linear function $f(x)$, where ϵ_k is the sum of the second order term in Taylor series expansion of the SGD+momentum operator with the stochastic noise induced by the SGD step at iteration k.

Given our analysis of the linear case, all perturbation results of (Scieur et al., 2016) and (Scieur et al., 2017) still apply. Assuming bounded perturbation in expectation, i.e.,

$$\mathbb{E}[\|\epsilon_k\|] \leq \nu \ \ \forall k$$

where ν typically bounds the deterministic perturbation arising due to the second order Taylor term and the variance of a SGD step, then, one can derive the global bound of Proposition 5.2 in (Scieur et al., 2017). This bound is difficult to analyze as it depends on the value of a so-called *regularized Chebyshev polynomial*, which is still unknown (but this time, it lies on the imaginary plane rather than the real line). However, it is possible

to give the relation between λ and ν to ensure the recovery of an optimal rate of convergence when $\varepsilon_k \to 0$. In particular, if

$$\lambda \in]\tau^{2/3}, \tau^0[, \qquad \tau = \frac{\nu}{\|x_0 - x^*\|},$$

then we asymptotically recover the rate in equation (6). The estimation of τ may be hard, but, in practice, to ensure good numerical convergence, it suffices to set $\lambda = O\left(\|R^T R\|\right)$, where the constant used is usually small (e.g. $\lambda = 10^{-8}\|R^T R\|$).

6.4 Numerical Range of Chambolle-Pock's Operator

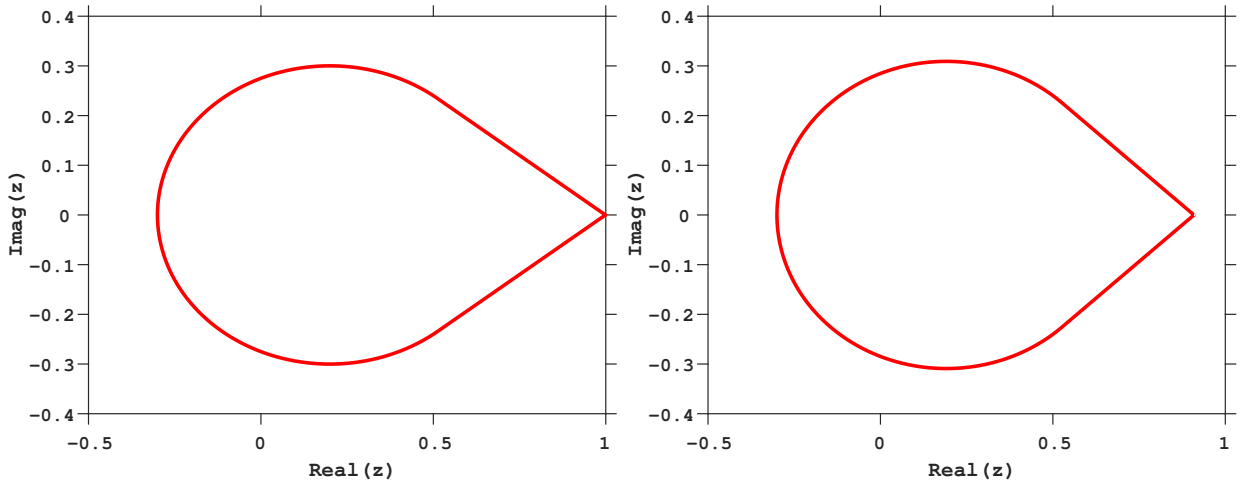

Figure 6: Field values for the Sonar dataset (Gorman and Sejnowski, 1988) with $\sigma = 4, \tau = 1/\|A^T A\|$. The dataset has been scaled such that $\|A^T A\| = 1$. Left: $\mu = 10^{-3}$, right: $\mu = 10^{-1}$. The smaller numerical range on the right means faster convergence.

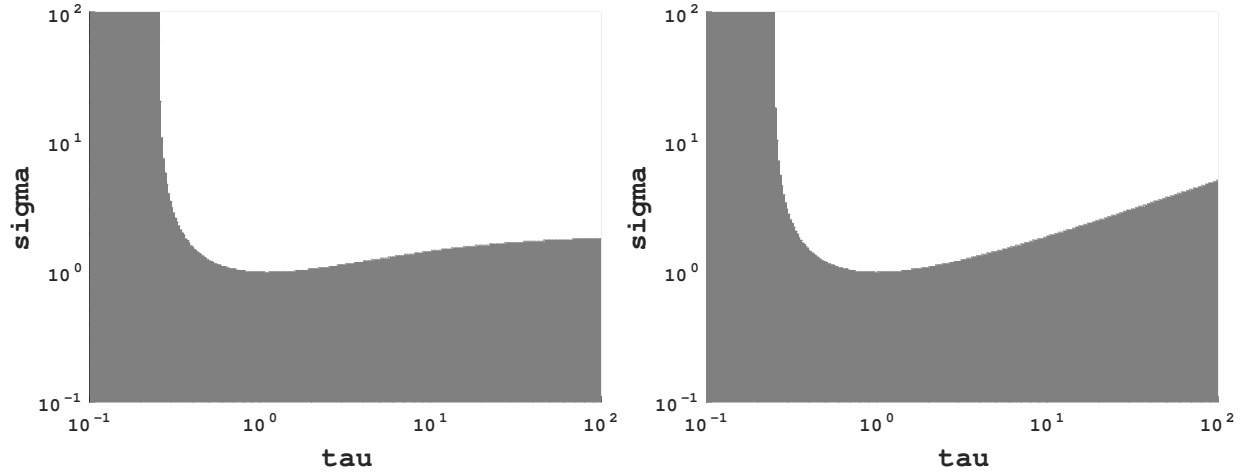

Figure 7: Plot of the $re(O^p) = 1$ frontier with degree $p = 5$ for the Sonar dataset (Gorman and Sejnowski, 1988) for different values of τ and σ. White color represents values for which $re(O^p) \leq 1$ (converging) and black color represents values $re(O^p) > 1$ (not converging). Left: $\mu = 10^{-3}$. Right: $\mu = 10^{-1}$.

6.5 Additional Numerical Results

6.5.1 Smooth Problems

We used binary classification datasets, available at UCI Machine learning repository[1], in the experiments.

Figure 8 shows the performance of different variants of the primal-dual algorithms on ridge regression problems for two different regularization constants on Sonar dataset. We observe that there is no significant difference in the performance of the method with the momentum term (θ) as compared to the one with no momentum term. We also observe that although the choice of the steplength parameters mentioned before have consistent performance across different problems, the improvements obtained with RNA are not very significant. However, choosing $\sigma = \tau = 1/\|A\|$ and applying RNA to the PDGM has consistently outperformed all other variants. This is in consistent with theoretical observations made in Section 4 that one can find optimal steplength parameters for which RNA is stable and obtains the optimal performance.

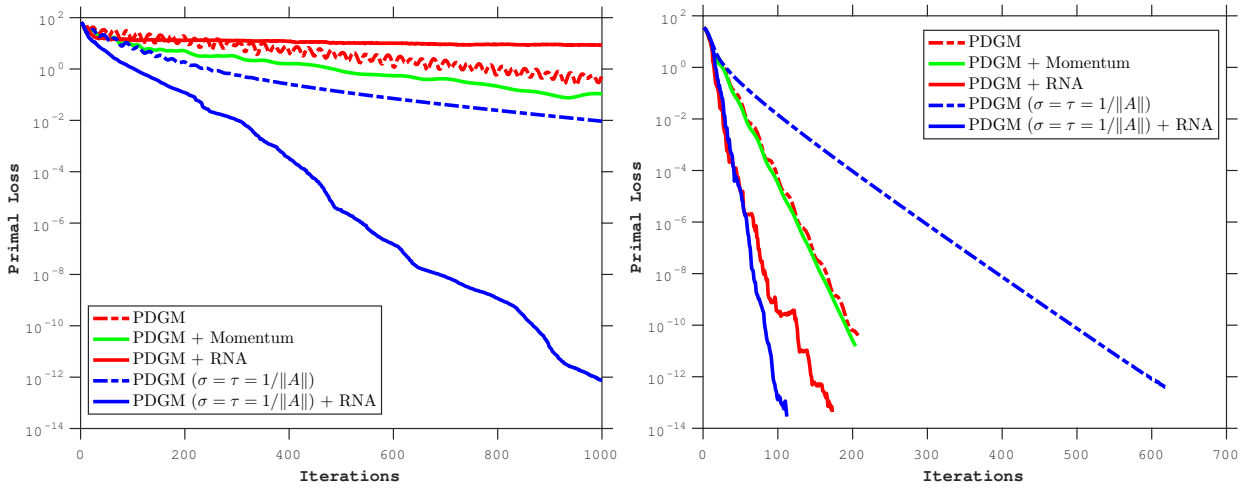

Figure 8: Quadratic loss on the Sonar dataset. Left : $\mu = 10^{-2}$. Right : $\mu = 10^{1}$. Comparison of online RNA with other variants of primal-dual gradient methods.

Figures 9, 10 and 11 compare the performance of primal-dual algorithms with other well know algorithms on ridge regression problems. We observe that Nesterov's accelerated gradient method and primal-dual gradient method consistently outperformed gradient descent as suggested by the theory as these methods achieve the optimal rates. The RNA variants of gradient descent and primal-dual methods are competitive and outperform their base algorithms.

[1] http://archive.ics.uci.edu/ml/index.php

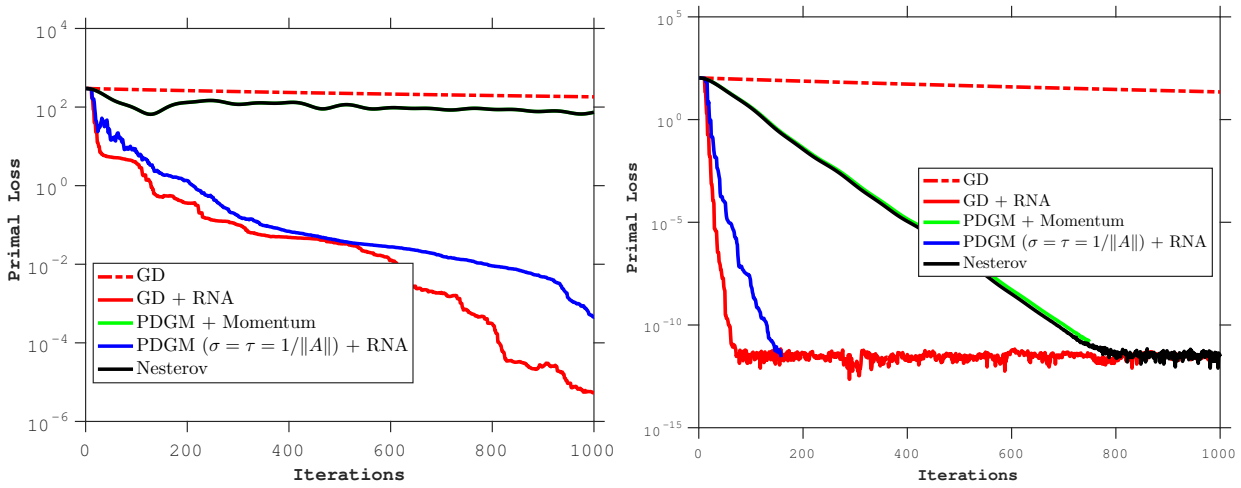

Figure 9: Quadratic loss on the Madelon dataset. Left : $\mu = 10^{-2}$. Right : $\mu = 10^{2}$. Comparison of online RNA on primal-dual gradient methods with other first-order algorithms.

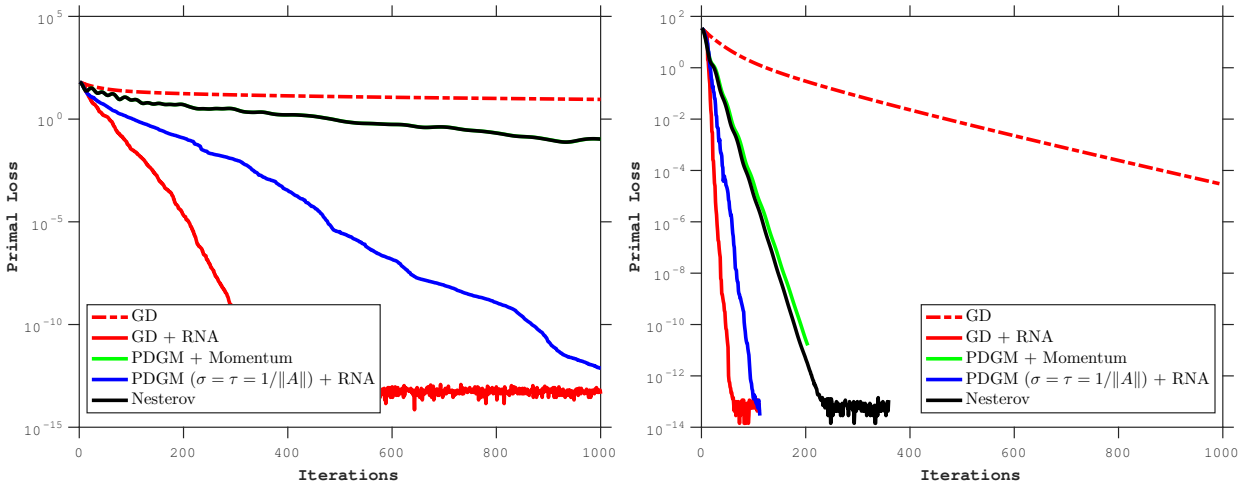

Figure 10: Quadratic loss on the Sonar dataset. Left : $\mu = 10^{-2}$. Right : $\mu = 10^{1}$. Comparison of online RNA on primal-dual gradient methods with other first-order algorithms.

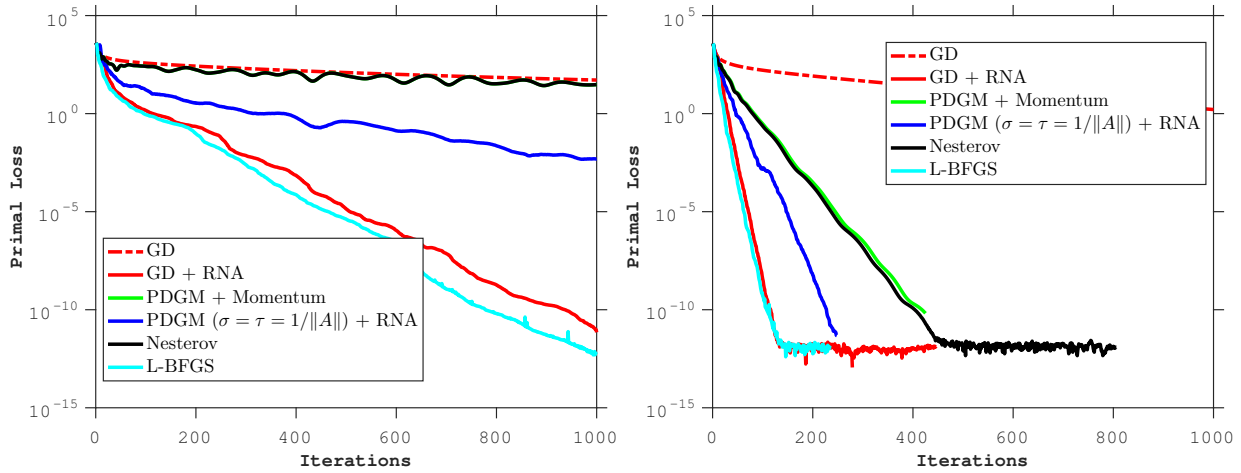

Figure 11: Quadratic loss on the Mushrooms dataset. Left : $\mu = 10^{-2}$. Right : $\mu = 10^{2}$. Comparison of online RNA on primal-dual gradient methods with other first-order algorithms.

Figures 12 and 13 show the performance of the methods on logistic regression problems on Madelon and Sonar datasets respectively. We observe that the RNA variants have substantially improved the performance of the base algorithms. The L-BFGS method with Armijo backtracking line-search has the optimal performance across different problems and the RNA variants are competitive to this method.

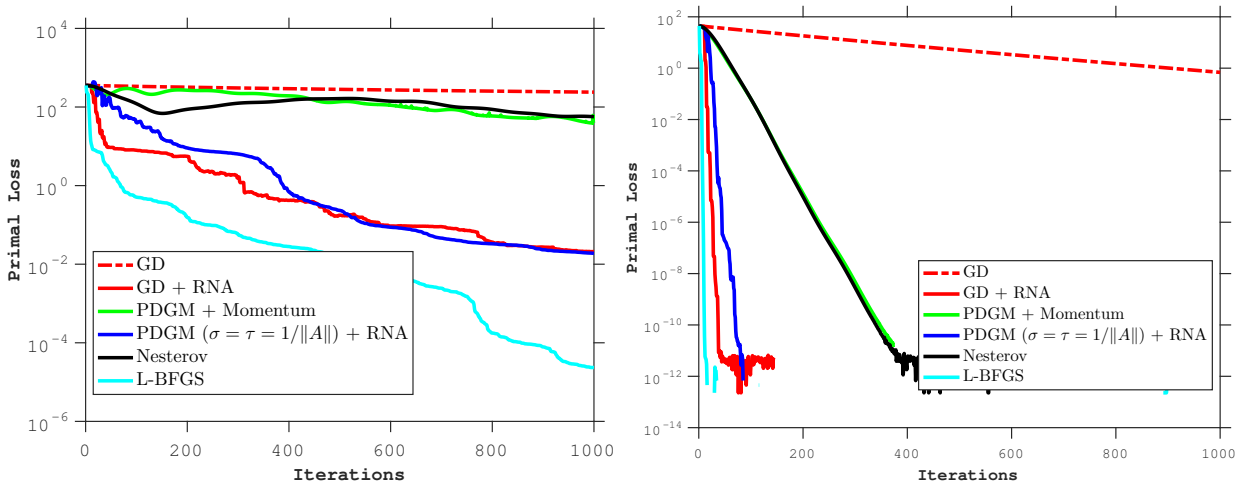

Figure 12: Logistic loss on the Madelon dataset. Left: $\mu = 10^{-2}$. Right: $\mu = 10^{2}$. Comparison of online RNA on primal-dual gradient methods with other first-order algorithms.

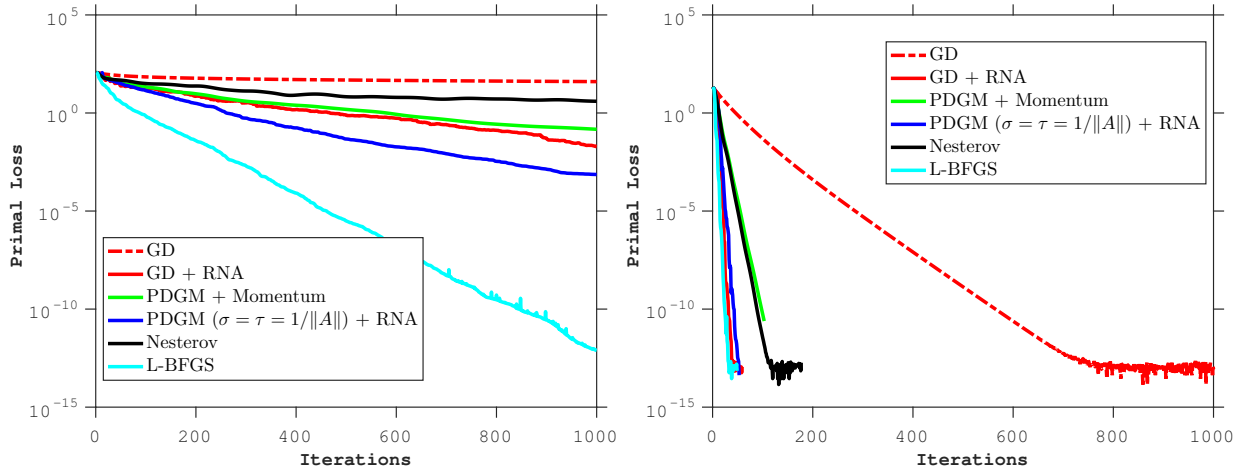

Figure 13: Logistic loss on the Sonar dataset. Left: $\mu = 10^{-3}$. Right: $\mu = 10^{1}$. Comparison of online RNA on primal-dual gradient methods with other first-order algorithms.

Figure 14 compares the performance of offline, restart and online versions of RNA on primal-dual gradient methods on the Sonar dataset. We observe that the improvement in the performance is more pronounced in the online version of RNA as compared to the offline version.

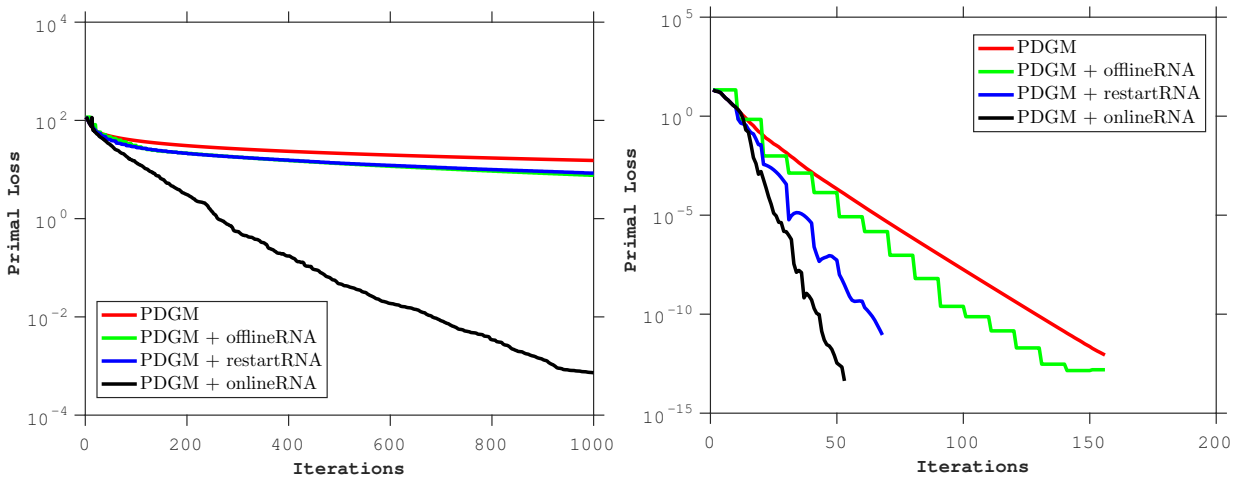

Figure 14: Logistic loss on the Sonar dataset. Left : $\mu = 10^{-3}$. Right : $\mu = 10^{1}$. Comparison of offline, restart and online variants of RNA on primal-dual gradient methods.

6.5.2 Non-Smooth Problems

Figure 15: Images used in the experiments. Left: True data. Middle: Noisy data with Gaussian noise $\zeta = 0.1$. Right: Noisy data with Gaussian noise $\zeta = 0.05$

Table 1 reports the number of iterations required for the distance between the primal function value and the optimal primal function value to be below certain accuracy. We observe that the PDGM + RNA has consistently outperformed the PDGM and its momentum variant for all accuracies.

	$\zeta = 0.1, \mu = 8$			$\zeta = 0.05, \mu = 16$		
	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$
PDGM	488	1842	7146	257	943	3706
PDGM + Momentum	377	1744	6813	226	921	3879
PDGM + offlineRNA	**221**	**1151**	**5801**	**141**	**671**	**3241**

Table 1: Number of iterations required for the primal accuracy to be below ϵ on the images shown in Figure 15 using primal-dual gradient methods.