Chapter 4

VAGA: Viscosity-based Accelerated Gradient Algorithm for Regularized Multitask Learning Framework with Convergence Guarantee and Applications

4.1 Introduction

In the field of machine learning, there exist various real world problems that involve multiple related subtasks of regression or classification. For example, in a spam detection system, the detection of spam e-mail/message in different languages can be considered as individual tasks. One approach for handling such a scenario is to consider each task individually under the framework of Single Task Learning, which does not exploit the relatedness of the tasks while learning. In a spam detection system, for example, all the tasks are related in terms of the common problem of spam classification. Patterns in spam or non-spam emails in the English language can help improve the classification process in other languages and vice versa. The main purpose of multitask learning (MTL) is to improve the generalization performance while learning multiple related tasks together. Various real-world applications of such algorithms include recognition [3, 183], recommender systems [5, 116], natural language processing [68, 60], computational biology [32, 223], web search ranking [51], online feature selection [210] etc.

In multitask learning framework, we utilize the internal relatedness between tasks during the process of learning so that the performance of individual tasks can be enhanced. Consider a supervised learning (regression) framework for $\mathscr{T}$ number of related tasks, where the training data set for t^{th} task is denoted by $\mathscr{D}_t = \{(x_{ti}, y_{ti}), x_{ti} \in \mathbb{R}^d, y_{ti} \in \mathbb{R}, i = 1, \cdots, m_t\}$ $\forall t = 1, \cdots, \mathscr{T}$. Here, the pair (x_{ti}, y_{ti}) represents i^{th} input/output pair of t^{th} task and m_t denotes the number of example pairs in t^{th} task. The set of $\{x_{ti}\}$ is denoted as X_t, the set of $\{y_{ti}\}$ is denoted as Y_t, and the parameter to learn is W. To learn a prediction function corresponding to each task $t \in \mathscr{T}$, is the main objective of multitask lasso problem, such that individual tasks can utilize the shared information between tasks. The minimization problem estimates parameter W from the training examples by solving the following,

$$\min_W \frac{1}{2} \sum_t \|X_t W_t - Y_t\|_F^2 + \rho \|W\|_{reg}, \tag{4.1}$$

where, $\|W\|_{reg}$ is a non-smooth regularizer term. We can consider both the ℓ_1 and ℓ_{21} norms for the framework. For ℓ_1 norm, $\|W\|_1$ is defined as the maximum absolute column sum of the matrix W, i.e., $\max\limits_{1 \le i \le t} \sum_{j=1}^d |w_{ij}|$, whereas for ℓ_{21} norm $\|W\|_{21}$ is defined as $\sum_{i=1}^d \sqrt{\sum_{j=1}^t |w_{ij}|^2}$, i.e., the sum of norm of each row. The $\|\cdot\|_{21}$ norm makes sure that W is sparse in rows, and selects features across tasks. In our experiments, we employed ℓ_{21} norm. Note that $X_t \in \mathbb{R}^{m_t \times d}$, $W_t \in \mathbb{R}^d$ and $Y_t \in \mathbb{R}^{m_t}$ corresponding to the t^{th} task. The ρ parameter is the sparsity controlling parameter and $\|\cdot\|_F$ is the Frobenius norm. The difference between single-task learning and multitask learning frameworks is illustrated in figure 4.1. There are t number of related tasks in the figure. In the single-task learning framework, the models are learned separately for individual tasks, thus not sharing any information while training. In the multitask learning framework, there is a joint training process for all the tasks such that all of them share the relatedness among them. Thus, the individual models we get after training carry the mutual information from other tasks as well.

A general notion to employ the task relatedness is to use a proper convex non-smooth regularization function. The overall formulation becomes a regularized risk minimization problem. Lasso framework is already discussed in Chapter 2. In multitask learning setting, various lasso extensions are available such as Group Lasso formulation [217] via the $l_1 - l_2$ block-norm [124] or the $l_1 - l_\infty$ block-norm [134, 196], tree structure [94], graph structure [4], flexible sparsity structure [107, 52] etc.

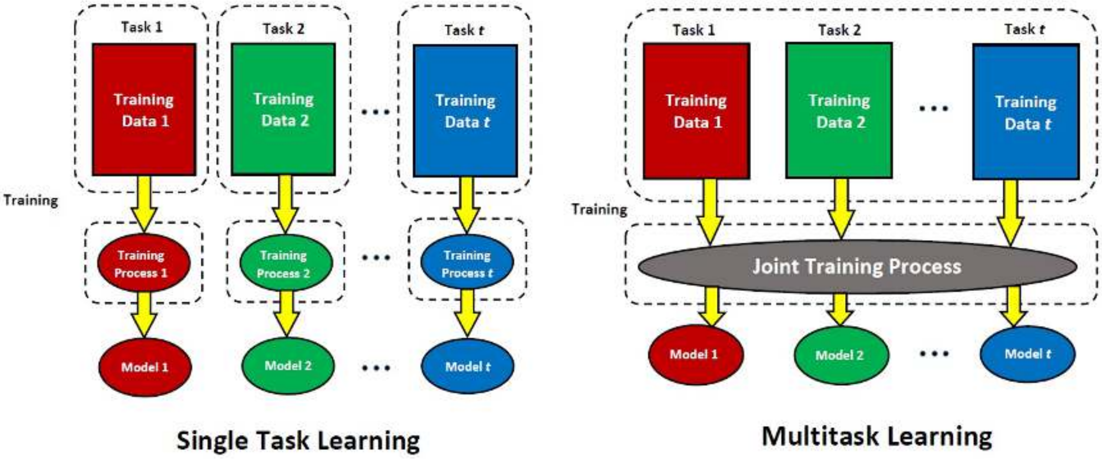


FIGURE 4.1: Single-task vs. Multitask Learning Frameworks

In convex optimization, proximal gradient-based methods are the methods of choice for solving MTL problems due to their applicability to large scale data. Previously proposed methods include interior point method [196], projected subgradient method [160], block-wise coordinate descent algorithm [121], forward-looking subgradients [77] and so on. Proximal methods, also called generalized gradient-based methods, are preferred over interior point methods (IPM) because of their computationally cheaper iterations. Although IPMs provide highly accurate results in low iteration counts, the iteration cost grows non-linearly with the number of decision variables [106]. The computationally cheaper iterations of First Order Methods (FOMs) motivated us to analyze them further under the framework of MTL. To improve the practical applicability of general proximal gradient techniques, often an inertial-based acceleration step is introduced that makes the process converge to the solution with $O(1/k^2)$ rate, where k is the number of iterations needed to reach the optimal point. The convergence proofs of such methods are either unavailable or exhibit a weak convergence in infinite dimensional space.

It has been stated in [90, 56], that the strong convergence of the sequence $\{x_n\}$ to a minimizer of function F, improves the convergence rate. In this direction, in [114] authors proposed the viscosity approximation method of selecting a particular fixed point of a given non-expansive mapping and proved the strong convergence in Hilbert spaces, which was further studied and extended in [208]. Recently, in [175] authors proposed a prox-Tikhonov like FBA under the framework of Banach spaces, considering the general condition of non-expansivity of operators and proved the strong convergence of the algorithm using viscosity approximation technique. This concept motivates us to analyze this and similar iterative schemes for the problem of regularized convex minimization problem and to investigate its practical performance on the multitask learning problems.

4.1.1 Contributions

The contribution reported in the chapter is three-fold:

- We apply the recent viscosity approximation based forward-backward algorithm [175] to solve the problem of multitask regression as viscosity-approximation-based proximal gradient algorithm (VPGA) and proposed a novel viscosity-approximation-based accelerated gradient algorithm (VAGA) for the problem of multitask regression.
- The boundedness of the sequence generated by the proposed algorithm and the strong convergence of VAGA is proved under specific conditions.
- The algorithm is applied to the problem of regularized multitask regression with sparsity-inducing regularizers. Experimental results are presented with three benchmark real datasets. VAGA is also applied to the popular bio-informatics problem of joint splice-site recognition and showed the performance with seven different genomes.

4.1.2 Outline

The rest of the chapter is organized as follows. Section 4.2 discusses the related mathematical concepts that are used in the proofs of theorems. We also give a brief discussion on the joint splice-site recognition problem in this section. In Section 4.3, we first apply the viscosity based proximal gradient algorithm (VPGA) to the regularized multitask learning problem. Next, we propose the novel viscosity-based accelerated gradient algorithm (VAGA) and apply this algorithm to the regularized multitask learning problem. In Section 4.4 we present the mathematical analysis of VAGA in terms of the boundedness and strong convergence of the sequence generated by this algorithm. Section 4.5 discusses the experimental setup and result analysis with several real datasets for both the problems of regularized multitask learning as well as joint splice-site recognition. We conclude this chapter in Section 4.6.

4.2 Preliminaries

In this section we will discuss a recent viscosity-approximation-based fixed point scheme, the related lemmas and theorems, we used in our proofs and the background of the Joint Splice-site Recognition task. The general framework we solve is as follows,

$$\min_{x} F(x) = g(x) + h(x). \tag{4.2}$$

As we know from the previous chapters, the proximal gradient methods [63] are a specific class of forward-backward splitting algorithms. To solve the general problem (4.2), a number of forward-backward splitting methods have been designed along with their convergence proofs such as [193, 64, 198, 123, 207], and references therein. Various advantages of forward-backward algorithms (FBA), such as convenience in applying, less costly iterations and good accuracy motivate researchers to further explore various modifications and generalizations of such methods. An attempt in this direction is to couple the classical FBA with the regularization/penalization methods, which in context of fixed-point theory can be considered as the viscosity-approximation. More discussion on the viscosity-approximation fixed-point schemes is given in the next section. The rich amount of research is already available in this direction, where traditional algorithms are combined with approximation methods such as [37, 8, 9, 114, 11] and references therein.

4.2.1 Viscosity-based Forward-backward Splitting Method (VFBA)

In [175], authors introduced the property $(\mathscr{N})$ for nonexpansivity of operators as follows. Let C be a nonempty closed convex subset of a Banach space X. An operator $B : C \to X$ is said to satisfy the property $(\mathscr{N})$ on $(0, \gamma_{X,B})$ if there exists $\gamma_{X,B} \in (0, \infty]$, depends on X and B, such that $I - \Psi B : C \to C$ is nonexpansive for each $\Psi \in (0, \gamma_{X,B})$. We directly utilize this property under the framework of Hilbert space. Let $\mathscr{C}$ be a non-empty closed convex subset of a Hilbert space $\mathscr{H}$. An operator $B : \mathscr{C} \to \mathscr{H}$ is said to satisfy the property $(\mathscr{N})$ on $(0, \gamma_{\mathscr{H},B})$ if there exists $\gamma_{\mathscr{H},B} \in (0, \infty]$, that depends on $\mathscr{H}$ and B, such that $I - \xi B : \mathscr{C} \to \mathscr{C}$ is non-expansive, for each $\xi \in (0, \gamma_{\mathscr{H},B})$.

Following Proposition 2.4 of [175], we can directly conclude that the forward-backward operator $J_{c_n}^{A,B}$ is non-expansive. Note that the fixed point of the mapping will correspond to the zero of $A+B$. Inspired by the fact that mapping $J_{c_n}^{A,B}$ is already split and a fixed point iterative algorithm for $J_{c_n}^{A,B}$ on $\mathscr{C}$ corresponds to a splitting algorithm for finding zeros of operator $(A+B)$, i.e., $0 \in J_{c_n}^{A,B}$, authors in [175] proposed the prox-Tikhonov type FBA under the framework of Banach space as follows:

$$x_{n+1} \leftarrow J_{c_n}^{A}(I - c_n B)((1-\beta_n)x_n + \beta_n f x_n), \quad \forall n \in \mathbb{N}, \tag{4.3}$$

where $f : \mathscr{C} \to \mathscr{C}$ is a contraction, $\{\beta_n\}$ is a sequence in $(0,1]$ and $\{c_n\}$ is a regularization sequence in $(0, \gamma_{\mathscr{H},B})$. With respect to the current context, we rename this algorithm as VFBA. In [175], authors proved the strong convergence of this algorithm in Banach space, under following parametric assumptions:

(A$_1$) $\lim\limits_{n\to\infty} \beta_n = 0$,
(A$_2$) $\sum_{n=1}^{\infty} \beta_n = \infty$,
(A$_3$) either $\sum_{n=1}^{\infty} |\beta_n - \beta_{n+1}| < \infty$ or $\lim\limits_{n\to\infty} |1 - \frac{\beta_n}{\beta_{n+1}}| = 0$,
(A$_4$) $0 < \varepsilon \le c_n$, where ε is a real number, and
(A$_5$) $\sum_{n=1}^{\infty} |c_n - c_{n+1}| < \infty, \quad \forall n \in \mathbb{N}$.

The following lemma from [208] is used in our results:

Lemma 4.1. *[208] Let (s_n) be a sequence of non-negative real numbers satisfying*

$$s_{n+1} \le (1-\theta_n)s_n + \theta_n \mu_n + \gamma_n, \quad n \ge 0, \tag{4.4}$$

where (θ_n), (μ_n) and (γ_n) satisfy the following conditions,
(i) $(\theta_n) \subset [0,1]$, $\sum_{n=1}^{\infty} \theta_n = \infty$, or equivalently $\Pi_{n=1}^{\infty}(1-\theta_n) = 0$, (ii) $\limsup_{n\to\infty} \mu_n \le 0$ and
(iii) $\gamma_n \ge 0 \ (n \ge 0)$, $\sum_n \gamma_n < \infty$.
Then, $\lim_{n\to\infty} s_n = 0$.

4.2.2 Joint Splice-site Recognition

One of the big challenges in the field of bioinformatics domain is to obtain the labeled dataset, which can be very costly. The concept of multitask learning, which incorporates information from several related tasks to improve the prediction accuracy, is useful in getting the information about unlabelled data. It is believed that combining the known information from several (related) species can help in predicting the information about the unknown species. The problem of computational biology we consider in this work is to predict the splice site in a DNA (Deoxyribonucleic Acid) sequence of eukaryote organisms. In the process of conversion of a DNA sequence to Protein, there are three sub-processes. These are (a) transcription (b) splicing and (c) translation. After the transcription process, we get a pre-mRNA (premature messenger Ribonucleic Acid) sequence generated from a DNA sequence. Such sequences are formed from basic nucleotide letters A (Adenine), C (Cytosine), G (Guanine) and T (Thymine). This pre-mRNA sequence consists of certain segments called introns and exons. The exon (coding region) segments provide useful information about the protein, whereas the intron (non-coding region) segment does not carry any useful information. Thus, it is necessary to locate the boundaries of exon segments, which together form the information carrier mRNA. The intron-exon boundary is called the acceptor site which has GT or GC letter sequence, while the exon-intron boundary is referred as the donor site, which has the AG letter sequence. However, not all locations where such letter sequences are present, are splice sites. The locations where these letter sequences are present but do not represent splice-sites are called decoy sites and are negative examples of the 2-class classification problem. Recognition task of such junctions helps in identifying whether such junctions are present in a pre-mRNA sequence. Figure 4.2 depicts the process in detail.

To improve the accuracy of the splice-site recognition task with multiple organisms, it seems to be a good idea to adapt or share the information among the organisms, which can be achieved through the multitask learning framework. With multitask learning framework, we can call this task as the joint splice-site recognition task. In the framework of multitask learning, each organism is a task. Each sequence is an instance in the training dataset and labels are positive and negative, i.e., each gene sequence is a training example, and if it contains a splice site, then it belongs to the positive class, otherwise to the negative class. The employed ℓ_{21} norm makes sure that W is sparse in rows, and is used to select only a few genes across tasks that are helpful in classifying the instances in positive or negative classes.

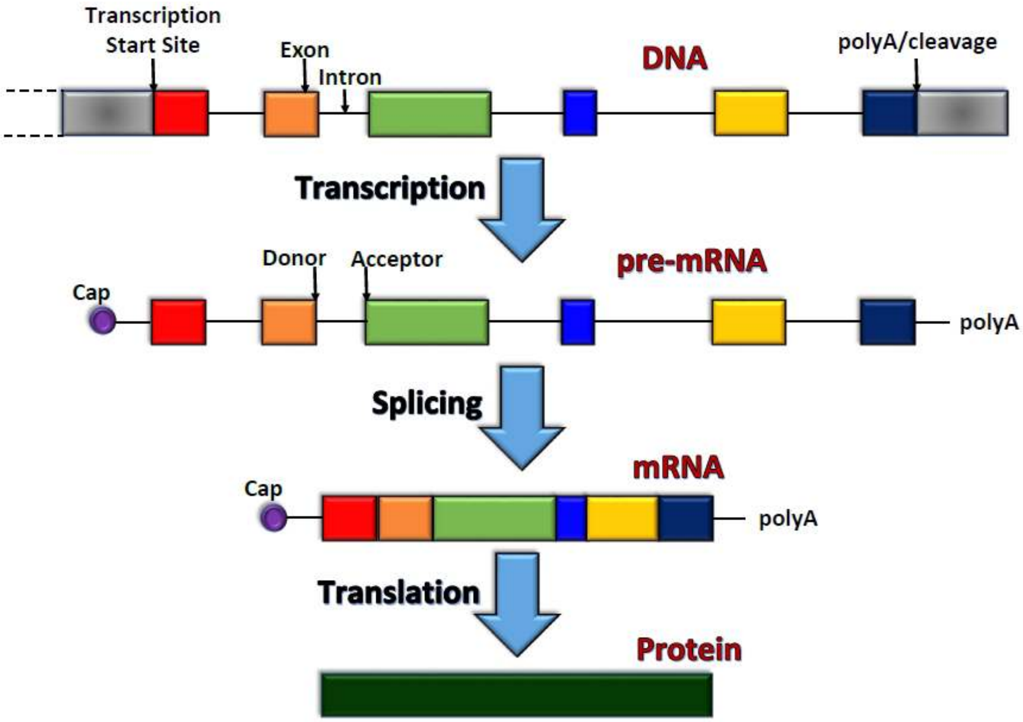


FIGURE 4.2: DNA to Protein Conversion. At 5' end (Left end), *cap* represents the added 7-methylguanosine, which we get after capping process. At the other end (3' end; right end), a poly(A) tail is formed with about 250 adenine residues.

4.3 VAGA

In this section, we first design the viscosity-based proximal algorithm for solving the multitask learning problem defined in (4.1). We next propose a novel viscosity-approximation-based inertial forward-backward algorithm in Hilbert space and the corresponding viscosity-based accelerated gradient algorithm for real finite-dimensional space.

Let $f : \mathscr{H} \to \mathscr{H}$ be a contraction mapping with contraction factor $\kappa \in [0,1)$, $\{\beta_n\}$ is a sequence in $(0,1]$ and $\{c_n\}$ is a regularization sequence in $(0, \gamma_{\mathscr{H},\nabla g})$. We solve problem (4.1) using the fixed point iteration given in (4.3) by defining the viscosity-based proximal gradient algorithm VPGA as follows.

$$\begin{aligned} P_n &\leftarrow (1-\beta_n)W_n + \beta_n f W_n, \\ W_{n+1} &\leftarrow J^{g,h}_{\rho c_n} P_n, \quad \forall n \in \mathbb{N}, \end{aligned} \tag{4.5}$$

where $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $(0,1]$ and the value of c_n is computed using the back-tracking method as in [23].

We first introduce the viscosity-based inertial Forward-Backward algorithm (VIFBA) for solving (4.2) and its strong convergence analysis in infinite-dimensional Hilbert space $\mathscr{H}$. Next we will apply it to solve problem (4.1). Consider $A : \mathscr{H} \to 2^{\mathscr{H}}$ and $B : \mathscr{H} \to \mathscr{H}$ as two maximal monotone operators, where B satisfies the property $(\mathscr{N})$ on $(0, \gamma_{\mathscr{H},B})$. Let $f : \mathscr{H} \to \mathscr{H}$ be a contraction mapping with contraction factor $\kappa \in [0,1)$, $\{\beta_n\}$ is a sequence in $(0,1]$ and $\{c_n\}$ is a regularization sequence in $(0, \gamma_{\mathscr{H},B})$. We define $T = J_c^A(I - cB)$ and $T_n = J_{c_n}^A(I - c_n B)$, where $\lim\limits_{n\to\infty} c_n = c$ for $c \in (0, \gamma_{\mathscr{H},B})$. We use $P_{\mathscr{C}}$ to denote the projection from $\mathscr{H}$ to $\mathscr{C}$. For any x_0 and $x_1 \in \mathscr{H}$, we define a new iterative scheme for sequence $\{x_n\}$ as follows:

$$\begin{aligned} z_n &\leftarrow x_n + \alpha_n(x_n - x_{n-1}), \\ y_n &\leftarrow (1-\beta_n) z_n + \beta_n f z_n, \\ x_{n+1} &\leftarrow T_n y_n, \quad \forall n \in \mathbb{N}, \end{aligned} \tag{4.6}$$

where $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $(0,1]$. The scheme defined by (4.6) will be called as viscosity-based inertial Forward-Backward Algorithm (VIFBA). This algorithm converges strongly to the solution point under some conditions discussed below. The term z_n is an inertial extrapolate step that produces acceleration with proper parameter settings and conditions on α_n. It should be noted that the term α_n is a generalized term, that was defined by the expression $\left(\frac{t_{n-1}-1}{t_n}\right)$ in [23] and $\frac{n-1}{n+3}$ in [47].

In order to solve the problem of multitask regression defined in (4.1), the problem set-up is same as defined in the previous section. The proposed algorithm is as follows,

$$\begin{aligned} P_n &\leftarrow W_n + \alpha_n(W_n - W_{n-1}), \\ Q_n &\leftarrow (1-\beta_n) P_n + \beta_n f P_n, \\ W_{n+1} &\leftarrow \operatorname{prox}_{\rho c_n \|\cdot\|_{21}}(Q_n - c_n(X^T X Q_n - X^T Y)). \end{aligned} \tag{4.7}$$

The pseudo-code of our iterative scheme VAGA (4.7) is shown in Algorithm 5. The behavior of the term α_n is shown in Chapter 2, which is useful in the proof of convergence for the proposed algorithm (4.7). In order to establish the convergence, we ensure that value of c_n belongs to set $(0, 2/L)$, where L is the Lipschitz constant of the gradient of the function $g(\cdot)$. In the large scale problems, since to compute the value of L is not easy, we are obtaining it from a line search technique opted from [23] in each iteration (as given in algorithm (5)). The initial value of c_n is set to 1.

```
Algorithm 5: VAGA
Data: Training/Testing Data, $\rho$, $tol$
Result: $W_{n+1}$
begin
    $W_0, W_1 \in \mathbb{R}^d$, $c_1 = 1$, $\alpha_1 = 0, n = 0$;
    repeat
        $n \leftarrow n + 1$;
        Find $c_n$ using backtracking step-size rule, and compute $\alpha_n$ and $\beta_n$;
        $P_n \leftarrow W_n + \alpha_n(W_n - W_{n-1})$;
        $Q_n \leftarrow (1-\beta_n)P_n + \beta_n f P_n$;
        $v_n \leftarrow (1-\beta_n)T_n + \beta_n u_n$;
        $W_{n+1} \leftarrow \text{prox}_{\rho c_n \|\cdot\|_{21}}(Q_n - c_n(X^T X Q_n - X^T Y))$;
    until converge;
```

The condition *converge* is considered to be achieved when the difference between the function value at the previous step and the function value at the current step becomes lesser than a previously defined tolerance value tol. In our experiments, we set the value of tol as 10e-5.

4.4 Analysis of VAGA

As described in previous section, the algorithm VAGA is based on more generalized VIFBA, in this section we discuss the boundedness and the strong convergence of the sequence generated by VIFBA (and hence for VAGA). In order to prove the convergence of the sequence $\{x_n\}$ defined by (4.6), in addition to the assumptions (A$_1$) - (A$_5$), we consider the following two conditions:
(C$_1$) $\sum_{n=1}^{\infty} \|x_{n+1} - x_n\|^2 < \infty$
(C_2) $\|x_n - x_{n-1}\|/\beta_n \to 0$ as $n \to \infty$.

We start our analysis with the boundedness proof of the sequence generated by the proposed algorithm.

4.4.1 Boundedness of $\{x_n\}$ from VAGA

Our next proposition proves that the sequence $\{x_n\}$ generated by (4.6) is bounded.

Proposition 4.2. *Let $\{x_n\}$ be a sequence in $\mathscr{H}$ generated by (4.6), where $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $(0,1]$ and $\{c_n\}$ is a regularization sequence in $(0,\gamma_{\mathscr{H},\,B})$ satisfying assumptions $(A_1)-(A_2)$, $(A_4)-(A_5)$ and condition (C_2). Then, the sequence $\{x_n\}$ is bounded.*

Proof. From the definition of y_n, we get,

$$\begin{aligned}\|y_n - x^*\| &\le (1-\beta_n)\|z_n - x^*\| + \beta_n\|fz_n - x^*\| \\ &\le (1-\beta_n)\|z_n - x^*\| + \beta_n(\|fz_n - fx^*\| + \|fx^* - x^*\|) \\ &\le (1-\beta_n)\|z_n - x^*\| + \beta_n(\kappa\|z_n - x^*\| + \|fx^* - x^*\|) \\ &= (1-(1-\kappa)\beta_n)\|z_n - x^*\| + \beta_n\|fx^* - x^*\|.\end{aligned}$$

Invoking (4.6), we have,

$$\begin{aligned}\|x_{n+1} - x^*\| = \|T_n y_n - x^*\| &\le \|y_n - x^*\| \\ &= (1-(1-\kappa)\beta_n)\|z_n - x^*\| + \beta_n\|fx^* - x^*\|.\end{aligned}$$

By condition $\frac{\|x_n - x_{n-1}\|}{\beta_n} \to 0$, there exists a constant $M \ge 0$ such that $\frac{\|x_n - x_{n-1}\|}{\beta_n} \le M$, $\forall n \in \mathbb{N}$. From the definition of z_n, we get

$$\begin{aligned}\|z_n - x^*\| \le \|x_n - x^*\| + \alpha_n\|x_n - x_{n-1}\| \;&=\; \|x_n - x^*\| + \alpha_n\frac{\|x_n - x_{n-1}\|}{\beta_n}\beta_n \\ \le \|x_n - x^*\| + \beta_n M, \quad \forall n \in \mathbb{N}.\end{aligned}$$

Substituting back, we get,

$$\begin{aligned}\|x_{n+1} - x^*\| &\le (1-(1-\kappa)\beta_n)(\|x_n - x^*\| + \beta_n M) + \beta_n\|fx^* - x^*\| \\ &\le (1-(1-\kappa)\beta_n)\|x_n - x^*\| + \beta_n(M + \|fx^* - x^*\|) \\ &\le \max\,\{\|x_n - x^*\|, \frac{M + \|fx^* - x^*\|}{(1-\kappa)}\} \\ &\;\vdots \\ &\le \max\,\{\|x_0 - x^*\|, \frac{M + \|fx^* - x^*\|}{(1-\kappa)}\}. \qquad (4.8)\end{aligned}$$

Hence the sequence $\{x_n\}$ is bounded. Also from (4.6), sequences $\{y_n\}$ and $\{z_n\}$ are also bounded. □

4.4.2 Convergence Analysis

In order to prove that the sequence $\{x_n\}$ generated by (4.6) is an approximating fixed point sequence of the operator T and thus, $\{y_n\}$ is an approximating fixed point sequence of T, we give the following proposition.

Proposition 4.3. *Let $\{x_n\}$ be a sequence in $\mathscr{H}$ generated by (4.6), where $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $(0,1]$ and $\{c_n\}$ is a regularization sequence in $(0,\gamma_{\mathscr{H},B})$ satisfying assumptions (A$_1$)-(A$_2$), (A$_4$)-(A$_5$) and condition (C$_2$). Then, $\|y_n - Ty_n\| \to 0$ as $n \to \infty$.*

Proof. Since $\{x_n\}$ is bounded and $\alpha_n \leq 1 \ \ \forall n \in \mathbb{N}$, from the definition of z_n we have,

$$\|z_n - x_n\| = \alpha_n\|x_n - x_{n-1}\| \leq \frac{\|x_n - x_{n-1}\|}{\beta_n}\beta_n.$$

Hence, from (C$_2$) we get $\|z_n - x_n\| \to 0$. Also, from (4.6), we can write

$$\|y_n - z_n\| = \beta_n\|z_n - fz_n\| \to 0 \text{ as } n \to \infty.$$

Again from (4.6), we have

$$\|x_{n+1} - T_nz_n\| = \|T_ny_n - T_nz_n\| \leq \|y_n - z_n\|.$$

From (4.6) and Proposition 2.2 from [175], for $K = \sup\limits_{n\in\mathbb{N}} \ \{\|Bx_n\| + \frac{1}{\varepsilon}\|(I - c_nB)x_n - J_c^A(I - c_nB)x_n\|\}$, we have

$$\begin{aligned}\|T_nz_n - Tx_n\| &\leq \|T_nz_n - T_nx_n\| + \|T_nx_n - Tx_n\| \\ &\leq \|z_n - x_n\| + |c_n - c|K.\end{aligned}$$

Finally, we have,

$$\begin{aligned}\|x_{n+1} - Tx_{n+1}\| &\leq \|x_{n+1} - T_nz_n\| + \|T_nz_n - Tx_n\| + \|Tx_n - Tx_{n+1}\| \\ &\leq \|y_n - z_n\| + \|z_n - x_n\| + |c_n - c|K + \|x_n - x_{n+1}\| \\ &= \|y_n - z_n\| + \|z_n - x_n\| + |c_n - c|K + \frac{\|x_n - x_{n+1}\|}{\beta_{n+1}}\beta_{n+1}\end{aligned}$$

Thus, from condition (C$_2$), we can conclude that sequence $\{x_n\}$ is an approximating fixed point sequence of T, i.e., $\|x_n - Tx_n\| \to 0$. Note that

$$\|y_n - x_n\| \leq (1-\beta_n)\|x_n - z_n\| + \beta_n\|x_n - fx_n\|.$$

Thus, $\|y_n - x_n\| \to 0$ as $n \to \infty$. It follows from Proposition 2.5 from [175] that $\{y_n\}$ is an approximating fixed point sequence of T. □

Now we are in the position to prove the strong convergence of the sequence $\{x_n\}_{n=1}^{\infty}$ generated by (4.6) with conditions (C$_1$)-(C$_2$).

Theorem 4.4. *Let $A : \mathscr{H} \to 2^{\mathscr{H}}$ and $B : \mathscr{H} \to \mathscr{H}$ are two maximal monotone operators such that zero set of $(A+B)$ is nonempty, and B satisfy the nonexpansivity property $(\mathscr{N})$ on $(0, \gamma_{\mathscr{H},B})$. Let $f : \mathscr{H} \to \mathscr{H}$ be a contraction operator with contraction parameter κ. For given $x_0, x_1 \in \mathscr{H}$, let $\{x_n\}$ be a sequence in $\mathscr{H}$ generated by (4.6), where $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $(0,1]$ and $\{c_n\}$ is a regularization sequence in$(0, \gamma_{\mathscr{H},B})$ satisfying assumptions (A$_1$)-(A$_2$), (A$_4$)-(A$_5$) and conditions (C$_1$)-(C$_2$). Then $\{x_n\}$ converges strongly to $x^* \in Zer(A+B)$, where $x^* = P_{Zer(A+B)}fx^*$.*

Proof. Since $P_{Zer(A+B)}f$ is a contraction, then there exists unique $x^* \in Zer(A+B)$ such that $P_{Zer(A+B)}fx^* = x^*$. Proposition 4.2 shows that sequence $\{x_n\}$ is bounded. Taking a suitable subsequence $\{x_{n_i}\}$ of $\{x_n\}$, we see that

$$\limsup_{n\to\infty}\langle fx^* - x^*, y_n - x^*\rangle = \lim_{i\to\infty}\langle fx^* - x^*, y_{n_i} - x^*\rangle. \tag{4.9}$$

We may assume that $x_{n_i} \rightharpoonup \hat{x}$ as $i \to \infty$. Note $||y_n - Ty_n|| \to 0$. From the demiclosed principle, we obtain that $\hat{x} \in Zer(A+B)$. From (4.9), the following statement can be justified following the variational inequality problem and Proposition 5.6.1 from [174]:

$$\begin{aligned}\limsup_{n\to\infty}\langle fx^* - x^*, y_n - x^*\rangle &= \lim_{i\to\infty}\langle fx^* - x^*, y_{n_i} - x^*\rangle \\ &= \langle fx^* - x^*, \hat{x} - x^*\rangle \leq 0.\end{aligned}$$

Observe that

$$\begin{aligned}\|x_{n+1}-x^*\|^2 &= \|T_ny_n - x^*\|^2 \leq \|(1-\beta_n)z_n + \beta_n fz_n - x^*\|^2\\ &= \|(1-\beta_n)(z_n-x^*) + (1-\beta_n)x^* + \beta_n(fz_n - fx^*) + \beta_n fx^* - x^*\|^2\\ &\leq \|(1-\beta_n)(z_n-x^*) + \beta_n(fz_n - fx^*)\|^2 + 2\beta_n\left\langle fx^* - x^*, y_n - x^*\right\rangle\\ &\leq (1-(1-\kappa)\beta_n)\|z_n - x^*\|^2 + 2\beta_n\left\langle fx^* - x^*, y_n - x^*\right\rangle\\ &= (1-(1-\kappa)\beta_n)\|x_n + \alpha_n(x_n - x_{n-1}) - x^*\|^2 + 2\beta_n\left\langle fx^* - x^*, y_n - x^*\right\rangle\\ &\leq (1-(1-\kappa)\beta_n)[\|x_n - x^*\|^2 + \alpha_n^2\|x_n - x_{n-1}\|^2\\ &\quad + 2\alpha_n\left\langle x_n - x^*, x_n - x_{n-1}\right\rangle] + 2\beta_n\left\langle fx^* - x^*, y_n - x^*\right\rangle.\end{aligned}$$

From Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}\|x_{n+1}-x^*\|^2 &\leq (1-(1-\kappa)\beta_n)[\ \|x_n - x^*\|^2 + \alpha_n^2\|x_n - x_{n-1}\|^2\\ &\quad + 2\alpha_n\|x_n - x^*\|\|x_n - x_{n-1}\|\ ] + 2\beta_n\left\langle fx^* - x^*, y_n - x^*\right\rangle\\ &\leq (1-(1-\kappa)\beta_n)\|x_n - x^*\|^2 +\ \|x_n - x_{n-1}\|^2\\ &\quad + 2\|x_n - x^*\|\|x_n - x_{n-1}\|\ + 2\beta_n\left\langle fx^* - x^*, y_n - x^*\right\rangle.\end{aligned}$$

Set $r = \max\ \{\|x_0 - x^*\|, \frac{M+\|fx^*-x^*\|}{(1-\kappa)}\}$. Hence from (4.8), $\{\|x_n - x^*\|\} \leq r\ \forall n = 0,1,\cdots$. Then we have,

$$\begin{aligned}\|x_{n+1}-x^*\|^2 &\leq (1-(1-\kappa)\beta_n)\|x_n - x^*\|^2 +\ \|x_n - x_{n-1}\|^2 + \frac{2r(1-\kappa)\beta_n\|x_n - x_{n-1}\|}{(1-\kappa)\beta_n}\\ &\quad + 2(1-\kappa)\beta_n\frac{\left\langle fx^* - x^*, y_n - x^*\right\rangle}{(1-\kappa)}.\end{aligned}$$

Set $\theta_n = (1-\kappa)\beta_n$, $s_n = \|x_n - x^*\|^2$, $\mu_n = 2r\|x_n - x_{n-1}\|/[(1-\kappa)\beta_n] + 2\left\langle fx^* - x^*, y_n - x^*\right\rangle/(1-\kappa)$ and $\gamma_n = \|x_n - x_{n-1}\|^2$. Then from condition (C$_1$) and lemma 4.1, we conclude that $\{x_n\}$ converges strongly to x^*. □

We give the following theorem for finding solution of (4.2). The operator $J_{c_n}^{A,B}$ is defined as the forward-backward operator $(I + c_n\partial h)^{-1}(I - c_n\nabla g)$. Hence, for any $c_n \in (0, 2/L)$, solutions of problem (4.2) are characterized by the fixed point equation,

$$x = prox_{c_n h}(I - c_n\nabla g)x. \tag{4.10}$$

We now apply Theorem 4.4 for finding solutions of non-smooth convex optimization problem (4.2).

Theorem 4.5. *Let $\mathscr{H}$ be a Hilbert space, $g : \mathscr{H} \to \mathbb{R}$ be a convex and differentiable function with an $L-$Lipschitz continuous gradient ∇g and $h : \mathscr{H} \to (-\infty, \infty]$ be a lower semi-continuous convex non-differentiable function with subgradient ∂h. For given $x_0, x_1 \in \mathscr{H}$, let $\{x_n\}$ be a sequence in $\mathscr{H}$ generated by*

$$\begin{aligned} z_n &\leftarrow x_n + \alpha_n(x_n - x_{n-1}), \\ y_n &\leftarrow (1-\beta_n)z_n + \beta_n f z_n, \\ x_{n+1} &\leftarrow prox_{c_n h}(I - c_n \nabla g)y_n, \quad \forall n \in \mathbb{N}, \end{aligned}$$

where $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $(0,1]$ and $\{c_n\}$ is a regularization sequence in $(0, 2/L)$ satisfying assumptions (A_1)-(A_2), (A_4)-(A_5) and conditions (C_1)-(C_2). Then $\{x_n\}$ converges strongly to x^*, where $x^* = P_{Zer(\nabla g + \partial h)} f x^*$.

Proof. Since ∇g satisfies property ($\mathscr{N}$) on $(0, 2/L)$, the result follows from theorem 4.4. □

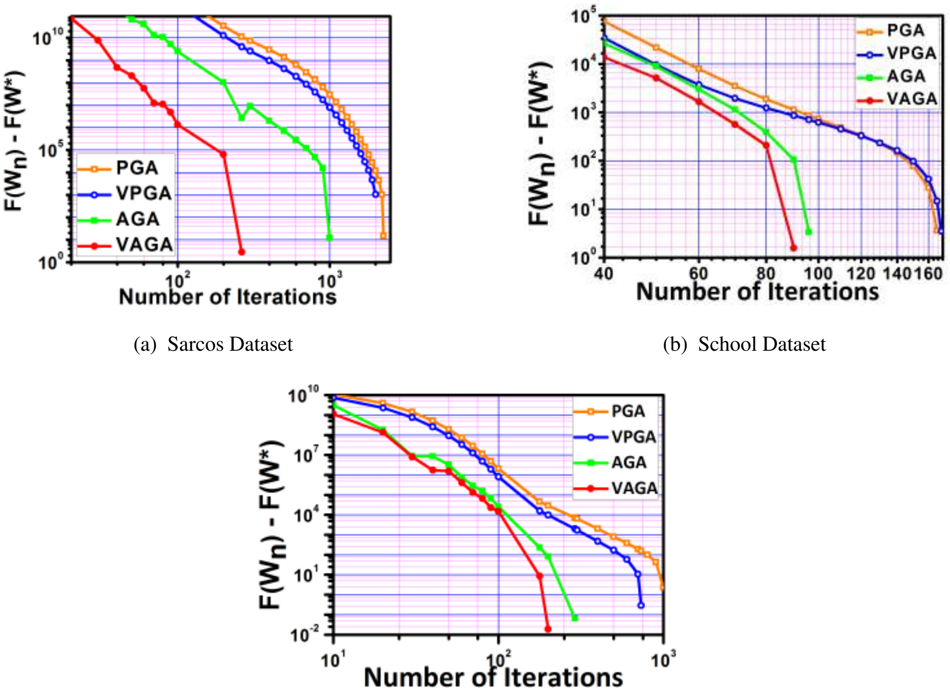
In the next section, we will describe the experimental set-up and the result analysis.

4.5 Experimental Results and Analysis

In this section, we present numerical experiments to demonstrate the performance of the proposed algorithm on multiple publicly available real datasets. All the experiments are performed on Intel core i7 processor with 10 GB RAM, under MATLAB computing environment. We have used MALSAR package [222] to design our experimental setup.

4.5.1 Multitask Regression

We compared our proposed algorithm with Proximal Gradient Algorithm (PGA) method, VPGA (4.5) and Accelerated Gradient Algorithm (AGA). For experiments, we employed three real multitask regression datasets as follows:



(a) Sarcos Dataset

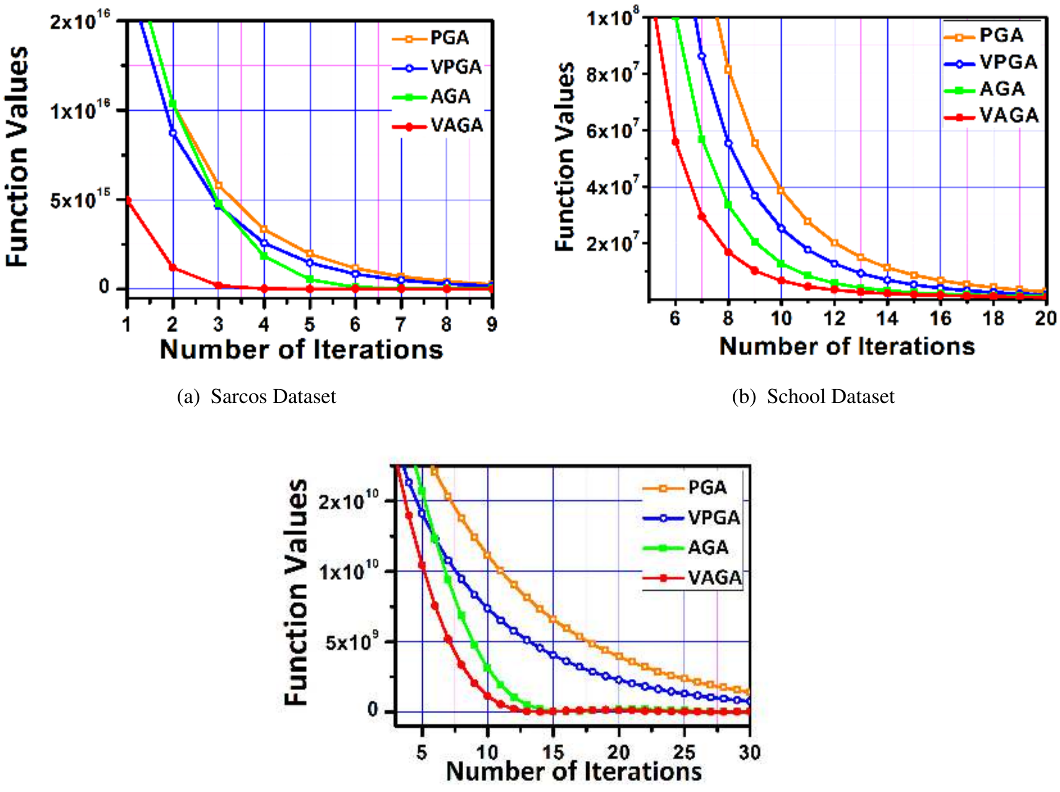
(b) School Dataset

(c) Parkinsons Dataset

FIGURE 4.3: Performance of VAGA on the basis of $F(W_n) - F(W^*)$ on the Sarcos, School and Parkinsons Datasets, respectively. $F(W_n)$ is the function value achieved after n^{th} iteration and $F(W^*)$ is the optimal function value. Shown graphs are log-log plots.

- **School:** Prediction of performance of students given their descriptions/record. The number of tasks is 139. In each task, there are 15362 examples with 28 dimensions.
- **Sarcos:** Prediction of inverse dynamics corresponding to the seven degrees-of-freedom of SARCOS anthropomorphic robot arm. The number of tasks is 51. In each task, there are 48933 examples with 21 dimensions.
- **Parkinsons:** Prediction of two Parkinson's disease symptom scores for patients based on bio-medical features. The number of tasks is 84. In each task, there are 5875 examples with 19 dimensions.

The value of sparsity controlling parameter ρ is set as $\theta \times \rho_{max}$, where ρ_{max} is set as $\|X^T Y\|_\infty$ and the value of θ is chosen from the set $\{$10, 5, 1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001$\}$. For the stopping criteria, the tolerance value tol (difference between two consecutive function values) is set to 10e-5, which also notifies the convergence. The maximum number of iteration is set to 10e4. All the vectors are initialized with a zero-valued vector. Values of c_n are initialized with 1.



(a) Sarcos Dataset

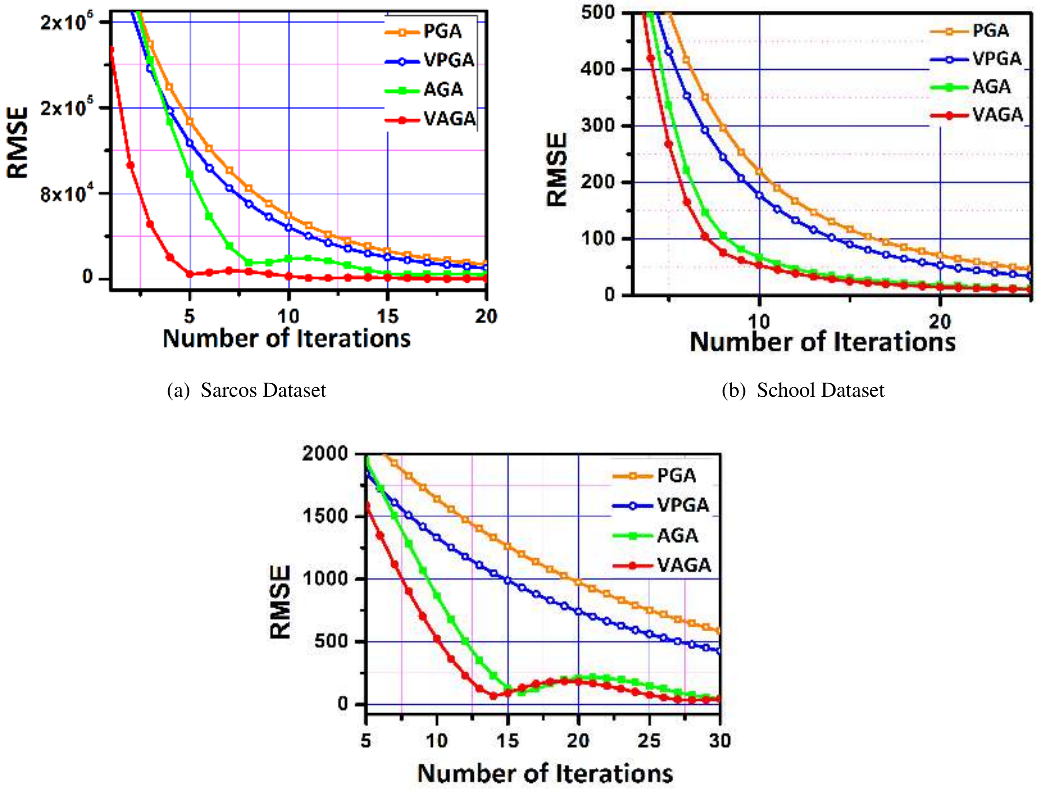
(b) School Dataset

(c) Parkinsons Dataset

FIGURE 4.4: Performance of VAGA on the basis of Reduction in Function Value in each Iteration for the Sarcos, School and Parkinsons Datasets.

There are few hyper-parameters based on which the convergence shown in the experiments depend. The first parameter is the parameter α_n at n^{th} iteration. The values of $\alpha_n \in (0,1]$ is set to satisfy the conditions given in previous section (4.7). It has been already known from [23, 47] that the definitions of α_n used in these works satisfy the condition of convergence. We set the parameter α_n as used in [47]. The second parameter is β_n, which is considered to belong to set $(0,1]$. We found in our experiments that for $\beta_n \rightarrow 1$ for VAGA, we achieve better convergence than the tradition AGA. We set β_n as $\frac{1}{(n+1)}$. The third parameter is the contraction parameter κ, which should belong to the range (0,1). As the initial investigation, we set the mapping $f(x)$ as a linear mapping $(\kappa \cdot x)$ with $\kappa = 0.8$.

To demonstrate the convergence of our algorithm, we performed experiments with 30% - 70% training and testing samples split for all the three datasets. The parameter ρ is tuned by five fold cross-validation for all methods. As a pre-processing step, z-score is



(a) Sarcos Dataset

(b) School Dataset

(c) Parkinsons Dataset

FIGURE 4.5: Performance of VAGA on the basis of rMSE Error rate for the Sarcos, School and Parkinsons Datasets.

performed on X_t, and a bias column is added to the data. As a performance measure, the standard root mean square error (rMSE) is used. Our first experiment demonstrates the convergence results of our algorithm on the three datasets. It can be observed from the figure (4.3) that the VAGA algorithm converges faster than AGA algorithm for all the three datasets. Graphs in figure (4.3) are the log-log plots, which holds the advantage to demonstrate the rate of convergence. As shown in figure (4.3), the slope of the curve of convergence of our algorithm gives the idea of its better convergence rate. However, the mathematical proof is the scope of future work.

In figure (4.4), we show the graph between the objective function value in each iteration. In each case, the function value achieved with VAGA is lesser than that of other algorithms. To compare the achieved accuracy, graphs are plotted between the rMSE and number of iterations, as shown in figure (4.5), which demonstrates that the rMSE values are decreasing with each iteration. For all the three datasets the rMSE value achieved is

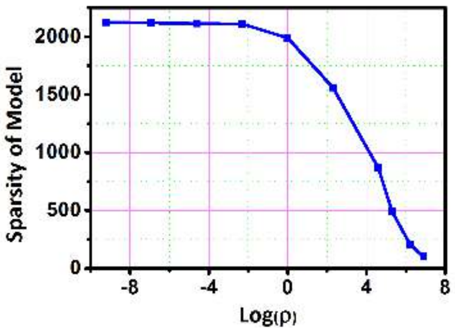
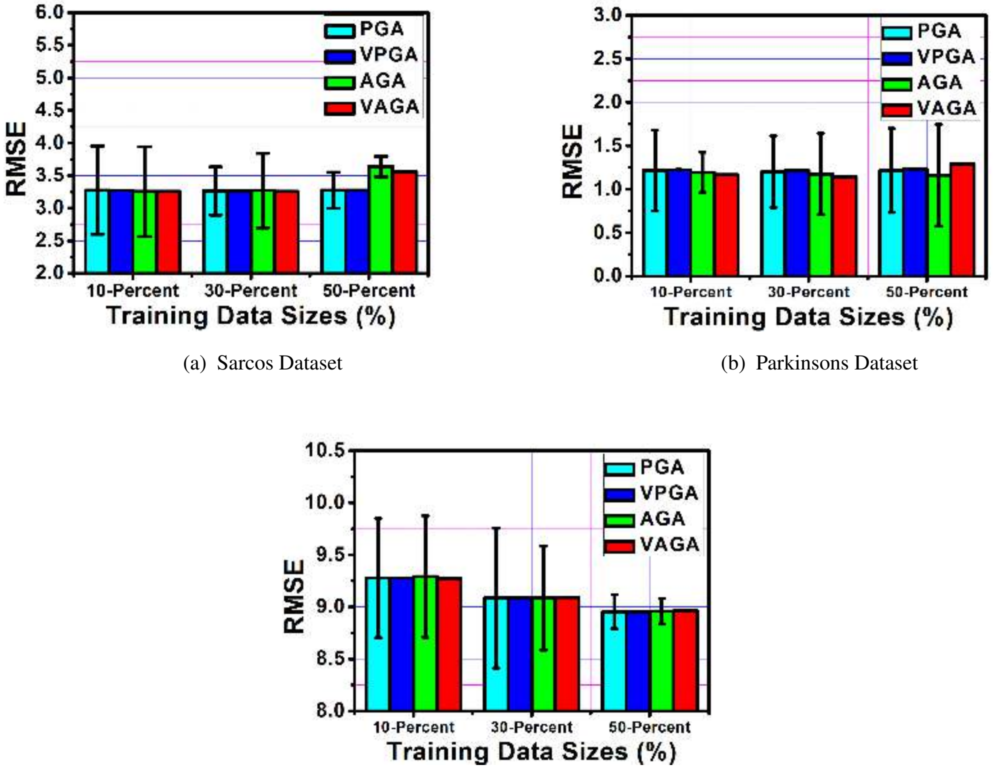


FIGURE 4.6: Sparsity of Predictive Model when Changing Regularization Parameter for the School dataset.

lesser than achieved by other algorithms. We also show the sparsity pattern in the learned parameter in (4.6) for School dataset with changing sparsity controlling parameter ρ. Similar trends were observed with the other two datasets. It is claimed in the field of fixed point theory that the viscosity-based approximation method, also called the regularized forward-backward algorithm, provides a stable solution. However, it has not been experimentally proved till date. Our next experiment shows the stability of the algorithm with respect to the previous traditional algorithms. One of the main and interesting observation is that with repeated experiments the variance obtained in the rMSE values is very less for the regularized algorithms, i.e., VPGA and VAGA, which represents the stability of the two algorithms. The stability results are shown in figure (4.7) for all the three datasets. For the PGA and AGA algorithms, the rMSE values highly vary for repeated experiments. For VPGA and VAGA, the quantity of variance is in range of (0.0001-0.0033). We repeated our experiments 30 times to get these variances in rMSE values. The number of iterations also vary, however, the quantity of this variance is not significant. In table 4.1, the number of iterations required to converge and root mean square error results, with training dataset size as 10%, 30% and 50% of the full datasets respectively are presented. Values in bold show the best result achieved on a dataset for each training set size. It can be observed from the table that for the Sarcos dataset, VAGA achieves the convergence in the least number of iterations for all the training set sizes. Also, it achieves the best accuracy (lowest error) for training set sizes 10% and 30%. The non-accelerated regularized prox-Tikhonov obtains the lowest error. However the number of iterations it consumes is much high. For Parkinson's datasets, VAGA again outperforms the rest of the algorithms and gives best results in the majority of cases. AGA achieves the lowest rMSE value for training set size 50%, whereas VAGA achieves the convergence fastest for all training set



(a) Sarcos Dataset

(b) Parkinsons Dataset

(c) School Dataset

FIGURE 4.7: Stability Results in terms of rMSE Values for all the three datasets.

sizes. With the School dataset, again VAGA beats all the algorithms in the majority of cases with the lesser number of iterations. However, here the best rMSE are obtained with PGA for 30% and 50% training set sizes.

4.5.2 Joint Splice-site Recognition

In this section, we present the experimental setup and result analysis for the task of joint splice site recognition with VAGA. We assumed each organism as a task. Since in this work, we are learning a model which recognizes whether the splice-sites are present or not in a pre-mRNA sequence, our model forms a binary classifier, which we designed as a logistic regression. The sparsity-inducing regularizer under consideration weights out those nucleotide letters, which are important in the joint splice-site recognition task. In order to learn splice sites using information among various species, we used genome data of six organisms as described below. We have used the SHOGUN 4.1.0 toolbox [182] in

TABLE 4.1: Results with no. of iterations required to reach the convergence and the final rMSE values with the Sarcos, School and Parkinsons Datasets on 10%, 30% and 50% Training Datasets respectively with repeated experiments.

		training data =10%		**training data =30%**		**training data =50%**	
		# Iter	**rMSE**	**# Iter**	**rMSE**	**# Iter**	**rMSE**
Sarcos	PGA	764	3.2745	1379	3.2659	1621	3.2764
	VPGA	652	3.2737	1217	3.2654	1457	**3.2756**
	AGA	314	3.2559	582	3.2696	720	3.6398
	VAGA	**263**	**3.2553**	**473**	**3.2561**	**329**	3.5612
Parkinsons	PGA	1759	1.2155	610	1.2031	513	1.2159
	VPGA	1463	1.2220	461	1.2118	386	1.2314
	AGA	263	1.1930	420	1.1745	355	**1.1615**
	VAGA	**206**	**1.1673**	**384**	**1.1445**	**257**	1.2915
School	PGA	395	9.2764	203	**9.0838**	113	**8.9541**
	VPGA	371	9.2769	188	9.0840	99	8.9553
	AGA	**118**	9.2923	95	9.0861	87	8.9597
	VAGA	119	**9.2743**	**88**	9.0896	**76**	8.9640

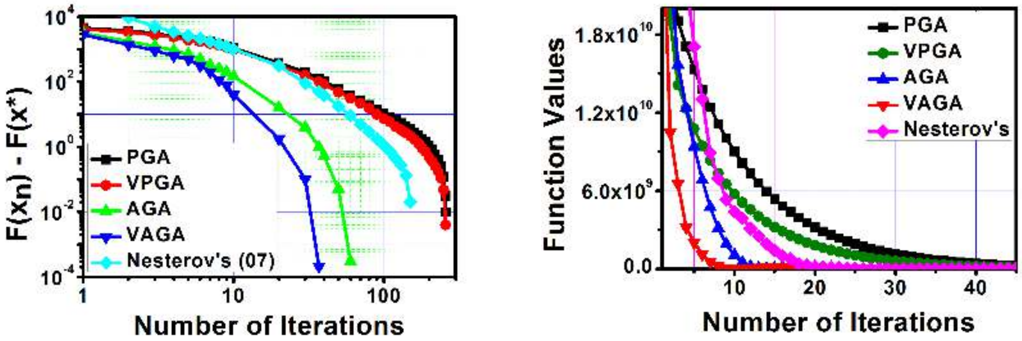
our experiments, which provides a wide range of libraries and algorithms applicable to computational biology domain. We also adapted few modules from the popular multitask learning toolbox MALSAR [222]. The genome datasets are obtained in the popular fasta format from ENSEMBL [99]. A short description of each considered organism is given in table 4.2. In addition to the previously compared algorithms (PGA, VPGA, AGA, VAGA), we also compare the performance of the proposed algorithm with the Nesterov's acceleration [139].

Our first result is the comparison of first-order algorithms by their convergence. Results are shown in figure 4.8(a) in terms of the log-log plot. It can be observed that the presented algorithm gives better convergence result than the previous state-of-the-art algorithms. In figure 4.8(b) we show the graph between the objective function value at each iteration. Also, with the VAGA algorithm, the function values reduce rapidly in comparison to that of other algorithms.

Finally, we check the performance of our algorithm in terms of area under the precision-recall curve (auPRC) for the joint splice site recognition task in figure 4.9. It can be observed from the figure that with VAGA the auPRC value is higher than the other algorithms for all the organisms. It should be noted that the presented results are only the initial investigation for the task of joint splice-site recognition. We aim to explore this field with more complex data structures and high-dimensional kernels.

TABLE 4.2: Summarization of the Datasets

Name	Description
	A. thaliana: Arabidopsis thaliana is a popular type of organism that is used in plant biology and genetics. It is known to have a small genome of approximately 135 Mb pairs, in the class of multicellular eukaryote.
	O. sativa: Oryza sativa is a scientific name of an Asian rice. The genome of this plant consists of a median total length of 359.938 Mb across 12 chromosomes.
	H. sapiens: Homo sapiens is the scientific name for the human species, which is the only surviving species of the genus Homo. The total length of the genome is 2996.42 Mb.
	M. musculus: Mus musculus or the house mouse is a small mammal, which is popular for modeling human disease and comparative genome analysis. The total length of the genome is 2671.82 Mb.
	O. latipes: Oryzias latipes, also known as medaka (rice fish) is an excellent model for genetic and environmental research. The genome is approximately 700Mb long across 24 pairs of chromosomes.
	D. rerio: Danio rerio, also known as Zebrafish is a small (4-5 cm) freshwater fish, which is widely used as a scientific model organism for the study of development biology and various human genetics diseases. The median total length of the genome is 1391.74 Mb.



(a) Performance of VAGA on the basis of $F(x_n) - F(x^*)$ for joint splice-site recognition task. Shown graphs are log-log plots.

(b) Performance of VAGA on the basis of Reduction in Function Value in each Iteration for the joint splice site recognition.

FIGURE 4.8: Results for the joint Splice-site Recognition task

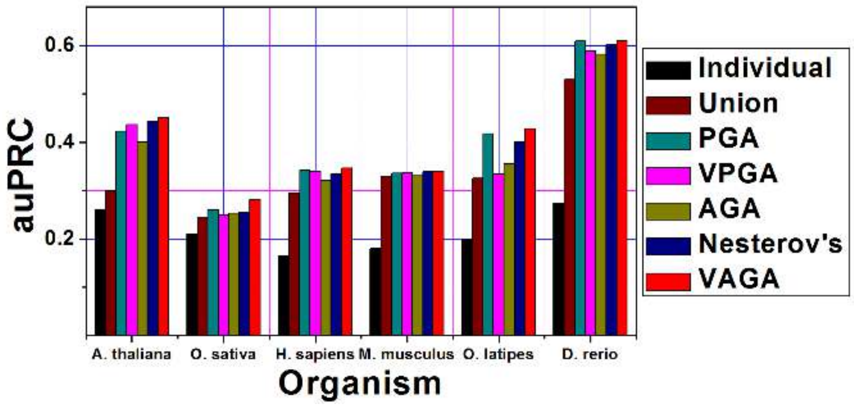


FIGURE 4.9: Results for the splice-site data sets from 6 eukaryotic genomes: Shown are auPRC performances of the few methods for each organism. We can observe that VAGA achieves the best results among all methods.

4.6 Conclusions

Two new viscosity-based proximal algorithms for the multitask learning problem, with sparsity-inducing regularizers are presented in this chapter. In addition to the introduction of a viscosity-approximation based proximal gradient algorithm, a novel viscosity-approximation based accelerated gradient algorithm is proposed, and the boundedness of the sequence generated by the algorithm is proved, We also proved that the sequence generated by the proposed algorithm converges strongly to a fixed-point solution. It is shown that the proposed algorithm converges faster to the traditional proximal-gradient algorithms for the regularized multitask regression problem. Also, we also applied the algorithm to the popular problem of joint splice-site recognition problem, where the multitask framework is applied to predict the splice-sites using genomes of six different organisms. The empirical results show that the proposed methodologies are faster as well as achieves good generalization on benchmark multitask learning datasets.