Published in final edited form as:
IEEE Int Conf Bioinform Biomed Workshops. 2015 November ; 2015: 1586–1590. doi:10.1109/BIBM.2015.7359912.

A Longitudinal Support Vector Regression for Prediction of ALS Score

Wei Du,
Center for Information Technology, National Institutes of Health, Bethesda, MD 20892-5624

Huey Cheung,
Center for Information Technology, National Institutes of Health, Bethesda, MD 20892-5624

Ilya Goldberg,
Intramural Research Program, National Institutes on Aging, Baltimore, MD 21224-6825

Madhav Thambisetty,
Intramural Research Program, National Institutes on Aging, Baltimore, MD 21224-6825

Kevin Becker, and
Intramural Research Program, National Institutes on Aging, Baltimore, MD 21224-6825

Calvin A. Johnson
Center for Information Technology, National Institutes of Health, Bethesda, MD 20892-5624

Calvin A. Johnson: johnson@mail.nih.gov

Abstract

Longitudinal studies play a key role in various fields, including epidemiology, clinical research, and genomic analysis. Currently, the most popular methods in longitudinal data analysis are model-driven regression approaches, which impose strong prior assumptions and are unable to scale to large problems in the manner of machine learning algorithms. In this work, we propose a novel longitudinal support vector regression (LSVR) algorithm that not only takes the advantage of one of the most popular machine learning methods, but also is able to model the temporal nature of longitudinal data by taking into account observational dependence within subjects. We test LSVR on publicly available data from the *DREAM-Phil Bowen ALS Prediction Prize4Life* challenge. Results suggest that LSVR is at a minimum competitive with favored machine learning methods and is able to outperform those methods in predicting ALS score one month in advance.

Keywords

longitudinal data; support vector regression; ALS; machine learning

I. Introduction

Longitudinal analyses are common in clinical research, particularly in longitudinal studies. These analyses are typically conducted via model-driven regression approaches such as linear regression models, mixed-effects models, or the generalized estimating equations [1,2]. These approaches assume a specific form of model (e.g., linear) and therefore require strong prior assumptions regarding the data. Furthermore, these methods require a primal-space implementation and are therefore not scalable to high-dimensionality data due to high computational demands.

The potential for machine learning techniques to make a contribution to longitudinal clinical studies was recently highlighted in a crowdsourcing competition known as the *DREAM-Phil Bowen ALS Prediction Prize4Life challenge* [3]. The goal of this challenge was to develop algorithms that can improve the prediction of Amyotrophic lateral sclerosis (ALS) (also known as Lou Gehrig's disease) progression as measured by the ALS Functional Rating Scale (ALSFRS). ALS is a fatal neurodegenerative disease with substantial heterogeneity in its clinical presentation. This makes diagnosis and effective treatment difficult. Surprisingly, none of the contestants explicitly modeled the temporal nature of the data in their training methods. Time-resolved features generally could not be incorporated into the machine-learning algorithms employed. Rather, participants performed linear regression on the time-varying features and represented those features by a slope and intercept [3]. (Other data reduction techniques were used as well, e.g., maximum/minimum representations.)

In another recent development, Chen and Bowman proposed a longitudinal support vector classifier (LSVC) as an approach that is scalable to classify high-dimensional longitudinal data such as neuroimaging data [4,5]. LSVC extends the well-known support vector machine (SVM) to longitudinal data by simultaneously estimating the traditional SVM separating hyperplane parameters with proposed temporal trend parameters. The authors provided only a limited test result on two time points of fMRI imaging data. To our knowledge, further extensive tests of LSVC have not been published.

We have hypothesized that Chen and Bowman's longitudinal extensions to the SVM could be further generalized as a longitudinal support vector regression (LSVR) and that the LSVR method may be applicable to longitudinal studies such as the ALS challenge. In this work, we present an evaluation of the LSVR on anonymized public data provided by the organizers of the DREAM-Phil Bowen ALS Prediction Prize4Life challenge. We did not seek to repeat the challenge itself. However, we did compare the performance of LSVR with an implementation of traditional linear support vector regression (SVR) [6] as well as a random forest approach, similar to the approach used by many of the challenge contestants. (e.g., [7]).

II. Methods

As LSVR is an extension of SVR, we first review the primal formulation of SVR and its dual form for quadratic programming (QP) optimization. Then we show how SVR is generalized to LSVR and describe the QP formulation of LSVR. Our derivation of LSVR closely follows Chen and Bowman's derivation of LSVM [4,5], including use of similar notation. The reader should be cautioned that implementation of LSVR requires a QP solver such as found in Matlab (The Mathworks, Natick, MA) or an original implementation of a QP solver. Standard SVM libraries such as libSVM [8] cannot be used with LSVR.

A. SVR and dual problem

Suppose we are given clinical training data $\mathbf{x_s} \in \mathscr{R}^p$ corresponding to N subjects, i.e., $s = 1, \ldots, N$ as well as N corresponding ALSFRS scores y_s. In ε-SVR, the goal is to find a function $f(\mathbf{x})$ that has at most ε deviation from the assessed scores y for all the training data, and at the same time is as flat as possible (equivalent to minimizing the gap in SVM) [9]. Then we have

$$f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + b.$$

We can write this problem as a convex optimization problem:

$$\min \qquad \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t. \qquad \begin{cases} y_i - \mathbf{w}^{\boldsymbol{T}}\Phi(\mathbf{x}_i) - b \le \varepsilon \\ \mathbf{w}^{\boldsymbol{T}}\Phi(\mathbf{x}_i) + b - y_i \le \varepsilon \end{cases}$$

where Φ is a function mapping **x** to the feature space. Analogously to the soft margin loss function that was adapted to SVM by Cortes and Vapnik [10], one can introduce slack variables ξ_i, ξ_i^* to the optimization problem. Hence we arrive at the formulation:

$$\min \qquad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^N (\xi_i + \xi_i^*)$$

$$\text{s. t.} \qquad \begin{cases} y_i - \mathbf{w}^{\boldsymbol{T}}\Phi(\mathbf{x}_i) - b & \le \varepsilon + \xi_i \\ \mathbf{w}^{\boldsymbol{T}}\Phi(\mathbf{x}_i) + b - y_i & \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \ge 0 \end{cases}$$

The constant C > 0 determines the trade-off between the flatness of f (which corresponds to the magnitude of $\|\mathbf{w}\|^2$) and the amount up to which deviations larger than ε are tolerated.

In most cases, it is computationally advantageous to solve the optimization problem in its dual formulation as described by:

$$\min_{\gamma,\gamma*} \frac{1}{2}\sum_{i,j=1}^N (\gamma_i - \gamma_i^*)(\gamma_j - \gamma_j^*)\mathbf{G}(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N y_i(\gamma_i - \gamma_i^*) + \varepsilon\sum_{i=1}^N (\gamma_i - \gamma_i^*)$$

$$\text{s.t.} \sum_{i=1}^N y_i(\gamma_i - \gamma_i^*) = 0 \text{and} \gamma_i, \gamma_i^* \in [0, C]$$

where $\mathbf{G}(\mathbf{x}_i, \mathbf{x}_j) =< \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) >$ acts as inner product that represents an entry in kernel matrix **G**. Then the above dual problem can be rewritten as

$$\min_{\gamma,\gamma^*}\frac{1}{2}[(\boldsymbol{\gamma}^*)^T,\boldsymbol{\gamma}^T]\left[\begin{array}{cc}\mathbf{G} & -\mathbf{G}\\ -\mathbf{G} & \mathbf{G}\end{array}\right]\left[\begin{array}{c}\boldsymbol{\gamma}^*\\ \boldsymbol{\gamma}\end{array}\right]+[\varepsilon\mathbf{e}^T-\mathbf{y}^T,\varepsilon\mathbf{e}^T+\mathbf{y}^T]\left[\begin{array}{c}\boldsymbol{\gamma}^*\\ \boldsymbol{\gamma}\end{array}\right]$$

$$\text{s.t.}\mathbf{z}^T\left[\begin{array}{c}\boldsymbol{\gamma}^*\\ \boldsymbol{\gamma}\end{array}\right]=0,\gamma_i,\gamma_i^*\in[0,C],i=1,2,\ldots,N$$

where $\mathbf{z}=[\underbrace{1,\ldots,1}_{N},\underbrace{-1,\ldots,-1}_{N}]^T$.

Once the separating hyperplane has been determined through QP optimization, the estimated regression value of a new observation $\mathbf{x}_i$ can be obtained as

$$\hat{y}_i=\sum_i\alpha_i\mathbf{G}(\mathbf{x}_i,\mathbf{x})+\hat{b}$$

where α is calculated by $\gamma^* - \gamma$ and nonzero α are considered as support vectors.

B. Longitudinal SVR

Consider longitudinal data collected from N subjects at T measurement occasion or visits, with p features quantified during each visit. The expanded feature matrix is then TN by p. Let $\mathbf{x}_s^{(t)}$ represent the features collected for subject s at time t. Hence, our aim is to assign each individual, a T-by-p matrix $\tilde{\mathbf{x}}_s=[\mathbf{x}_s^{(1)},\mathbf{x}_s^{(2)},\ldots,\mathbf{x}_s^{(T)}]^T$, a T-by-1 vector $\tilde{\mathbf{y}}_s=[y_s^{(1)},y_s^{(2)},\ldots,y_s^{(T)}]^T$. We predict linear trends of change

$$\mathbf{x}_s^{(1)}+\beta_1\mathbf{x}_s^{(2)}+\ldots+\beta_{T-1}\mathbf{x}_s^{(T)}=\tilde{\mathbf{x}}_s^T\boldsymbol{\beta}$$

$$y_s^{(1)}+\beta_1 y_s^{(2)}+\ldots+\beta_{T-1}y_s^{(T)}=\tilde{\mathbf{y}}_s^T\boldsymbol{\beta}$$

where unknown parameter vector $\boldsymbol{\beta} = [1, \beta_1, \beta_2, \ldots, \beta_{T-1}]^T$ is a T-by-1 vector. The trend information takes into account observational dependence within subjects. We intend to jointly estimate the parameter vector β and α in a LSVR model.

The Lagrangian function incorporating longitudinal parameter is as follows:

$$\begin{aligned}\mathscr{L} := \frac{1}{2}\mathbf{w}^T\mathbf{w}+C\textstyle\sum_{i=1}^{N}(\xi_i+\xi_i^*) - \sum_{i=1}^{N}(\eta_i\xi_i+\eta_i^*\xi_i^*)\\ -\sum_{i=1}^{N}\gamma_i(\varepsilon+\xi_i - y_i+<\mathbf{w},\mathbf{x}_{\boldsymbol{i}}>+b)\\ -\sum_{i=1}^{N}\gamma_i^*(\varepsilon+\xi_i^*+y_i - <\mathbf{w},\mathbf{x}_{\boldsymbol{i}}> - b)\\ =\frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{N}(\xi_i+\xi_i^*) - \textstyle\sum_{i=1}^{N}(\eta_i\xi_i+\eta_i^*\xi_i^*)\\ -\sum_{i=1}^{N}\gamma_i(\varepsilon+\xi_i - \tilde{y}_i^T\beta+<\mathbf{w},\tilde{\mathbf{x}}_i^T\beta>+b)\\ -\sum_{i=1}^{N}\gamma_i^*(\varepsilon+\xi_i^*+\tilde{y}_i^T\beta - <\mathbf{w},\tilde{\mathbf{x}}_i^T\beta> - b)\end{aligned} \quad (1)$$

Here $\mathscr{L}$ is the Lagrangian and η_i, η_i^*, γ_i, γ_i^* are the Lagrange multipliers. It follows from the saddle point condition that the partial derivatives of $\mathscr{L}$ with respect to the primal variables ($\mathbf{w}$, b, ξ_i, ξ_i^*) have to vanish to optimize the Lagrangian function. Thus we are left with

$$\frac{\partial\mathscr{L}}{\partial b}=\sum_{i=1}^{N}(\gamma_i^* - \gamma_i)=0 \quad (2)$$

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}}=\mathbf{w} - \sum_{i=1}^{N}(\gamma_i - \gamma_i^*)\tilde{\mathbf{x}}_i^T\beta=0 \quad (3)$$

$$\frac{\partial\mathscr{L}}{\partial\xi_i}=c - \gamma_i - \eta_i=0 \quad (4)$$

$$\frac{\partial\mathscr{L}}{\partial\xi_i^*}=c - \gamma_i^* - \eta_i^*=0 \quad (5)$$

Substituting equations (2) through (5) into (1) yields the dual-space quadratic programming problem:

$$\max \qquad -\frac{1}{2}\textstyle\sum_{i,j=1}^{N}(\gamma_i-\gamma_i^*)(\gamma_j-\gamma_j^*)\boldsymbol{\beta}^{\boldsymbol{T}}\tilde{\mathbf{x}}_{\boldsymbol{i}}\tilde{\mathbf{x}}_j^T\boldsymbol{\beta} +\sum_{i=1}^{N}\tilde{y}_i^T\boldsymbol{\beta}(\gamma_i-\gamma_i^*)-\varepsilon\sum_{i=1}^{N}(\gamma_i-\gamma_i^*)$$

$$\text{s. t.} \qquad \textstyle\sum_{i=1}^{N}(\gamma_i^* - \gamma_i)=0\text{and}\gamma_i, \gamma_i^* \in [0, C]$$

Similarly, the above dual problem can be rewritten as

$$\min_{\boldsymbol{\alpha}_m} \frac{1}{2}\boldsymbol{\alpha}_m \left[\begin{array}{cc} \mathbf{G}_m & -\mathbf{G}_m \\ -\mathbf{G}_m & \mathbf{G}_m \end{array} \right] \boldsymbol{\alpha}_m + \mathbf{y}_m^T \boldsymbol{\alpha}_m$$

where $\alpha_m = [\gamma^{*T} \beta_1 \gamma^{*T} \dots \beta_{T-1} \gamma^{*T} \gamma^T \beta_1 \gamma^T \cdots \beta_{T-1} \gamma^T]^T$, $\mathbf{y}_m = [\varepsilon \mathbf{e}^T - y^{(1)T} - y^{(2)T} \dots - y^{(T)T}$ $\varepsilon \mathbf{e}^T + y^{(1)T} y^{(2)T} \cdots y^{(T)T}]^T$ and

$$\mathbf{G}_m = \left[\begin{array}{ccc} \mathbf{G}^{(1,1)} & \dots & \mathbf{G}^{(1,T)} \\ \vdots & \ddots & \vdots \\ \mathbf{G}^{(T,1)} & \dots & \mathbf{G}^{(T,T)} \end{array} \right],$$

subject to

$$\mathbf{z}^T \left[\begin{array}{c} \gamma^* \\ \gamma \end{array} \right] = 0, \gamma_i \gamma_i^* \in [0, C], i = 1, 2, \dots, N$$

and there is no constraint on β.

Parameters α_m can be determined using QP and then β can be estimated from α_m to obtain the relationship among responses of different time points.

III. Experimental Results

We investigated the performance of the proposed method by applying it to public ALS challenge data and comparing the results with that of two popular machine-learning methods, specifically LibLinear SVR [11] and the Random Forest (RF) implementation function *TreeBagger* in Matlab.

A. Data set

Experimental data were downloaded from The DREAM-Phil Bowen ALS Prediction Prize4Life challenge website. This large data set comprises 1824 anonymized patients from phase 2 and 3 clinical trials. Although the original challenge segmented the data into training, validation, and holdout sets, we rather followed a cross-validation protocol. Up to twelve months of longitudinal data as well as the corresponding ALSFRS are included in the feature space. There are about 44 time varying features, including protein biomarkers, urine pH, calcium, etc., as well as 34 constant features, including age, race, sex, etc. In our tests, instead of estimating the slope between months 3 and 12 (as was done in the original challenge), we predict the next month's ALSFR based on the clinical data up to and including the current month.

B. Experimental design

Let $y_s^{(t)}$ represents the ALSFRS of month t for subject s. When predicting $\hat{y}_s^{(t+k)}$, the training features include clinical data up to month t as well as $y_s^{(1)}, y_s^{(2)}, \dots, y_s^{(t)}$. Then we predict

$\hat{y}_s^{(t+k)}$ by training obtained static features for another LibLinear and RF, renamed as LibLinear-M and RF-M. In this paper, we predict ALSFRS one, two, and three months ahead, e.g., $\hat{y}_s^{(t+k)}$, $k = 1,2,3$, using data up to month t. When $k = 1$, we investigate the prediction performance of using month 1 to t ($t = 3,4, \dots 11$), respectively. When $k = 2$ and $k = 3$, we evaluate the performance of up to month 10 and 9, respectively.

We perform the same experiments using LSVR, LibLinear SVR and RF. LibLinear SVR and RF were each implemented with two treatments of the data in the feature vectors. For those algorithms, in order to provide versions of the tests comparable with the ALS challenge addition, we convert each type of time-resolved features per patient into static features by calculating their mean across t months. These reduced-data implementations are referred to as LibLinear-M and RF-M in the results. We also implement LibLinear and RF feature vectors which were constructed by concatenating the features from each time point; these implementations are simply referred to as LibLinear and RF.

During the cross-validation, we randomly select training and testing subjects from the available data set, and calculate the root-mean square error (RMSE) for each algorithm. This procedure repeats 100 times, where each run is an independent trial, to obtain the average RMSE. The algorithm parameters used in LSVR and LibLinear are the same, namely linear kernel with box constrain as 10. RF is implemented as an ensemble of 50 trees.

C. Results

We vary the number of training data from 50 to 150 subjects, incrementing upward by 25 subjects per test. The number of testing data is fixed at 50 subjects. In general, we find that both LSVR and RF respond to increased training data through better regression performance all the way up to 150 subjects and likely beyond. LibLinear SVM performance, however, improves little beyond 100 subjects in training.

Fig. 1 shows the prediction error of the various algorithms trained with 100 and 150 training subjects to predict ALSFRS for the next month. It indicates that LSVR yields the best performance of the various algorithms when predicting ALSFRS at 6 months and beyond. Using more time points to predict the next one leads to slightly better results for the three algorithms. The mean-feature implementations of LibLinear-M and RF-M are not competitive after month 5. The incorporation of additional data points seems to deteriorate the performance of LibLinear-M and RF-M. Like LSVR, the “stacked feature” implementations LibLinear and RF display a similar oscillatory pattern after month 5.

Results of predicting ALSFRS two and three months ahead are shown in Fig. 2. We can observe that the performance of all algorithms worsens as the prediction interval increases. RF appears to have gained the advantage predicting three months with LSVR remaining competitive. The trend of the curve suggests that LSVR’s performance in predicting beyond one month ahead improves with additional temporal data.

IV. Discussion and Conculusions

In this paper, we propose a novel longitudinal machine learning algorithm that takes into account observation dependence within subjects through estimation of additional weighted parameters corresponding to the different time points. It allows simultaneous estimation of the SVR hyperplane parameters and the temporal trend parameters. Our derivation of LSVR was motivated by the apparent need for methods that fuse longitudinal modeling and machine learning paradigms. We drew heavily from the work of Chen and Bowman [4,5], extending their longitudinal SVM classifier to perform regression. While Chen and Bowman's work was foundational and influential, their testing of their longitudinal classifier was rather restricted to two samples. In addition to extending the longitudinal support vector classifier to regression, we sought to gain a more thorough understanding of the performance of LSVR by testing it under various conditions and by training and testing it with different amounts of data.

The publicly available ALS challenge data on which we tested LSVR serve as a surrogate for a wide class of clinical longitudinal data in which both temporal data (including blood biomarkers) and constant data (e.g., demographic) data are available. Our results suggest that LSVR is indeed competitive with currently favored machine-learning methods. In testing LSVR against implementations of a random forest and a conventional SVR, LSVR seems to find its forte in predicting ALS score one time-point (one month) ahead. LSVR also appears to respond favorably to increased training data and, to a lesser extent, to multiple time points tested.

At this point, we are unable stipulate on the extent to which the comparative performance of LSVR with the other algorithms is a function of the nature of the ALS data, or whether we can expect to see similar trends in other data. Quite simply, more testing is required on additional data sets. However, the results do suggest that LSVR is worthy of such further tests. A common problem in longitudinal clinical data is missing data. Unfortunately, LSVR does not model missing data, although the effect of estimating missing data appears to be no more severe than in other algorithms. We postulate that LSVR may find its most suitable application in prediction of high-dimensional time-series genomic data, since LSVR is amenable to parallel implementation and the computational performance of LSVR is certainly scalable.

ACKNOWLEDGMENT

This research was supported by the CIT and NIA Intramural Research Programs.

References

1. Lewsom, JTN.; Jones, RN.; Hofer, SM. Longitudinal data analysis: A practical guide for researchers in aging, health and social sciences. London, UK: Routledge; 2012.
2. Locascio JJ, Atri A. An overview of longitudinal data analysis methods for neurological research. Dement Geriatr Cogn Dis Extra. 2011; 1(1):330–357. [PubMed: 22203825]
3. Kuffner R, Zach N, et al. Crowdsourced analysis of clinical trial data to predict amyotrophic lateral sclerosis progression. Nat Biotechnol. 2015; 1:51–57. [PubMed: 25362243]

4. Chen S, Bowman FD. A novel support vector classifier for longitudinal high-dimensional data and its application to neuroimaging data. Statistical Analy Data Minng. 2011; 4:604–611.

5. Chen S, Grant E, Wu TT, Bowman FD. Some recent statistical learning methods for longitudinal high-dimensional data. Computatinal Statistics. 2014; 6:10–18.

6. Drucker, H.; Burges, CJC.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. In: Mozer, MC.; Jordan, MI.; Petsche, T., editors. Advances in Neural Information Processing Systems. Vol. 9. Cambridge, MA: MIT Press; 1997. p. 155-161.

7. Hothorn T, Jung HH. RandomForest4Life: a Random Forest for predicing ALS disease progression. Amyotroph Lateral Scler Frontotemporal Degener. 2014; 15(5–6):444–452. [PubMed: 25141076]

8. Chang C-C, Lin C-J. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2011; 2:27. 1–27:27.

9. Smola AJ, Scholkopf B. A Tutorial on Support Vector Regression. Statistics and Comuting. 2004; 14:199–222.

10. Cortes C, Vapnik V. Support vector networks. Machine Learning. 1995; 23:273–297.

11. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research. 2008; 9:1871–1874.

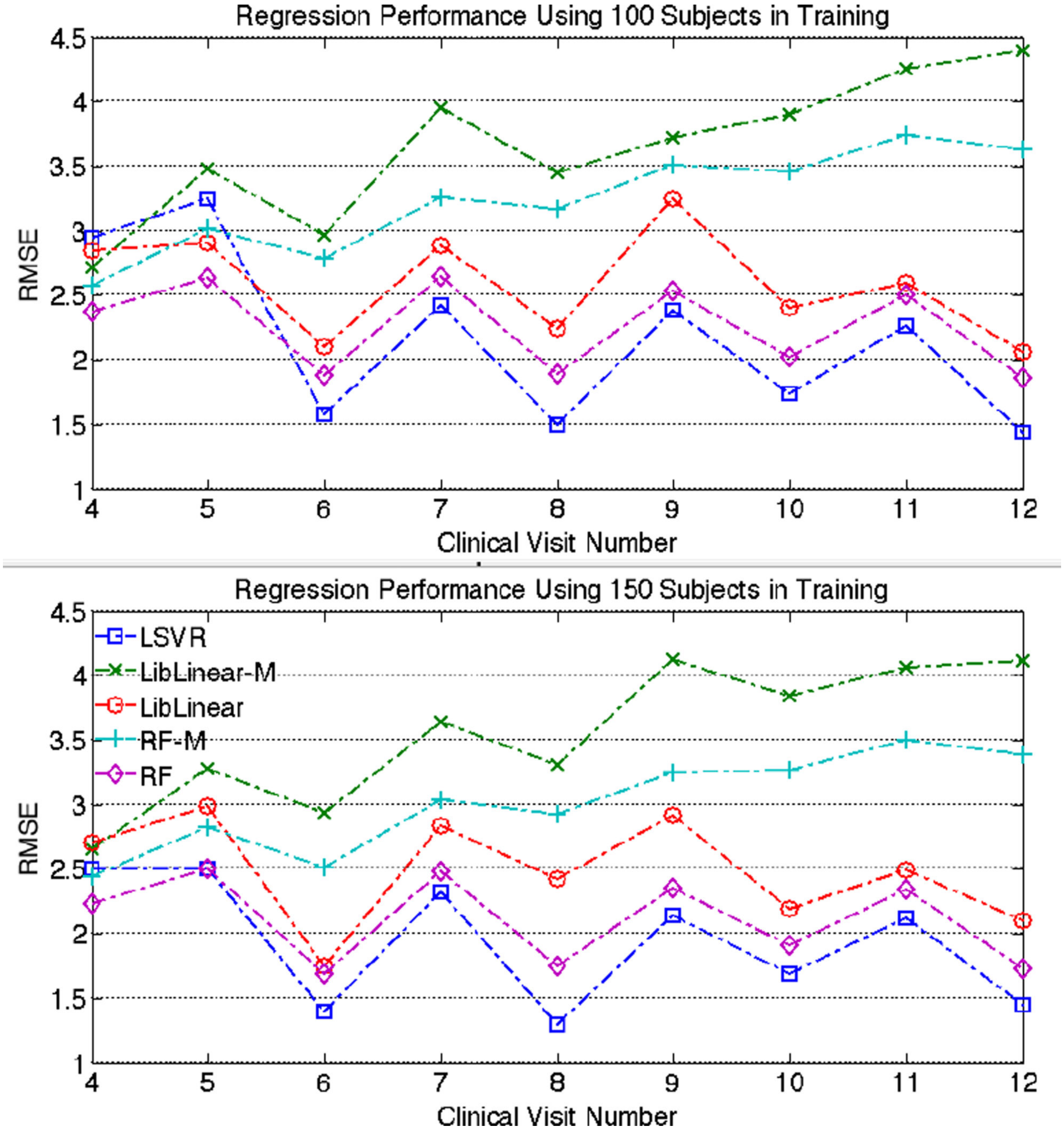

Fig. 1.

Performance comparison when predicting ALSFRS of the next month. The numbers of subjects in training are 100 and 150. The x-asis represents which month's ALSFRS is predicted. The y-axis denotes the average RMSE. Smaller RMSE indicates better regression performance. LSVR yields the best performance amongst the algorithms when estimating ALSFRS after month 5th.

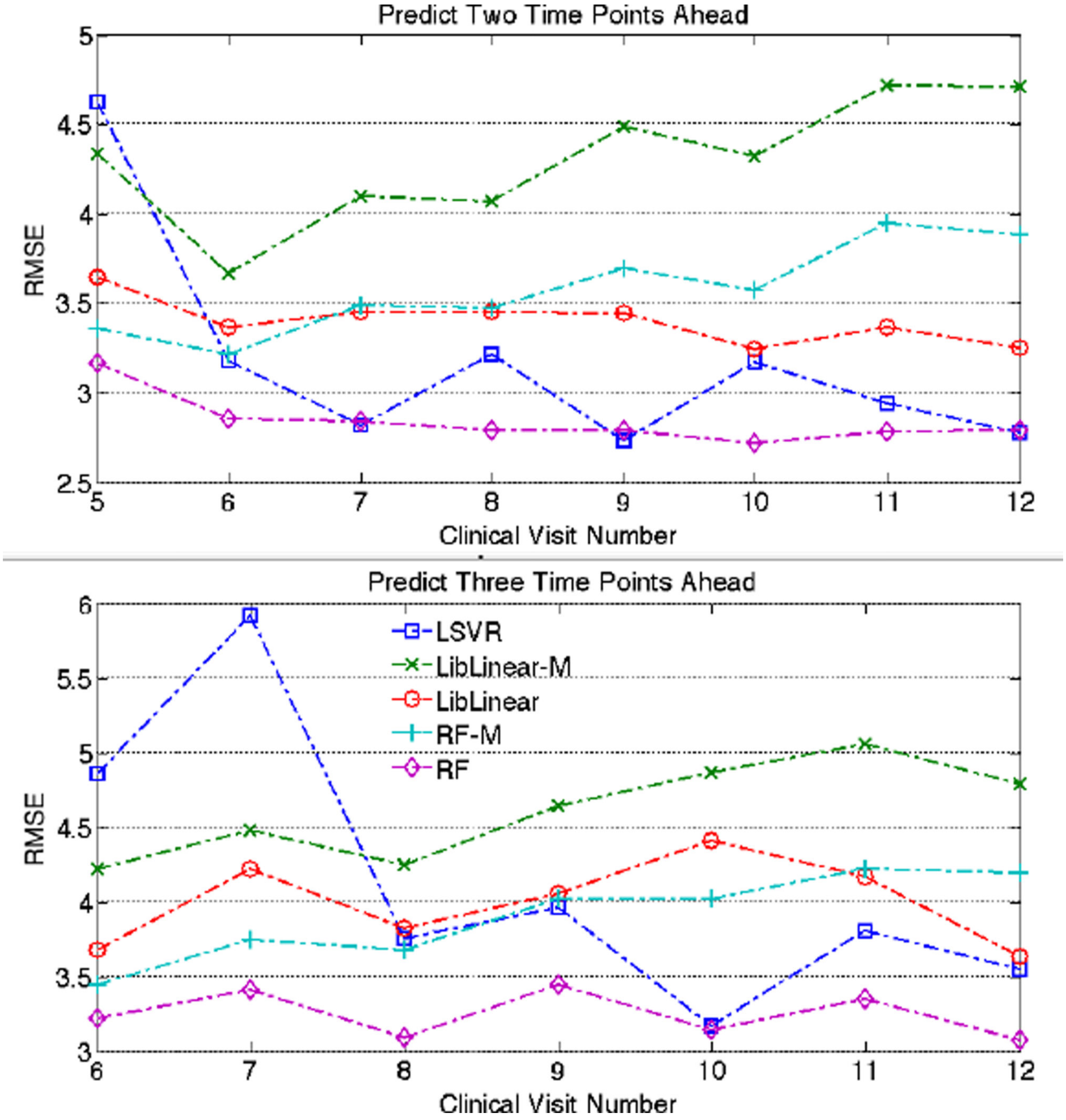

Fig. 2.

Performance comparison between LSVR and the other algorithms when prediction ALSFRS two and three months ahead. The horizontal axis indicates number of monthly clinical visit.