A Sparse Bayesian Learning Algorithm for Longitudinal Image Data

Mert R. Sabuncu[*]

A.A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital,
Harvard Medical School, Charlestown, MA, USA

Abstract. Longitudinal imaging studies, where serial (multiple) scans are collected on each individual, are becoming increasingly widespread. The field of machine learning has in general neglected the longitudinal design, since many algorithms are built on the assumption that each datapoint is an independent sample. Thus, the application of general purpose machine learning tools to longitudinal image data can be sub-optimal. Here, we present a novel machine learning algorithm designed to handle longitudinal image datasets. Our approach builds on a sparse Bayesian image-based prediction algorithm. Our empirical results demonstrate that the proposed method can offer a significant boost in prediction performance with longitudinal clinical data.

Keywords: Machine learning, Image-based prediction, Longitudinal data.

1 Introduction

Machine learning algorithms are increasingly applied to biomedical image data for a range of clinical applications, including computer aided detection/diagnosis (CAD) and studying group differences, e.g. [1,2,3] . In early biomedical applications, off-the-shelf algorithms such as Support Vector Machines were employed on image intensity data. However, there has been a recent proliferation of customized methods that derive optimal image features and incorporate domain knowledge about the clinical context and imaging data, e.g. [4,5,6,7,8]. Such customized methods can offer a significant increase in prediction accuracy.

Machine learning in general, and its application to population-level biomedical image analysis in particular, has largely been concerned with the cross-sectional design, where each sample is treated as independent. Yet, as data acquisition costs continue to fall and data collection efforts become more collaborative and standardized, longitudinal designs have become increasingly widespread. Longitudinal studies, where serial data are collected on each individual, can offer increased sensitivity and specificity in detecting associations, and provide insights into the temporal dynamics of underlying biological processes.

Real-life longitudinal data suffer from several technical issues, which make their analysis challenging. Subject drop-outs, missing visits, variable number of

[*] Supported by NIH NIBIB 1K25EB013649-01 and a BrightFocus grant (AHAF-A2012333). Data used were obtained from ADNI: http://tinyurl.com/ADNI-main.

N. Navab et al. (Eds.): MICCAI 2015, Part III, LNCS 9351, pp. 411–418, 2015.
DOI: 10.1007/978-3-319-24574-4_49

Table 1. Data from annual ADNI MRI visits analyzed in this study. Note some subjects had MRI visits at 6, 18, 30, 42, 54, and 66 months too.

Planned visit time (months)	Baseline	12	24	36	48	60	72
Mean± Std. time (months)	0	13.1 ± .8	25.5 ± 1.2	37.7 ± 1.2	50.7 ± 2.2	62.4 ± 1.7	74.2 ± 2.0
Number of imaging sessions	791	649	518	336	216	159	131

visits, and heterogeneity in the timing of visits are commonplace. For example, Table 1 illustrates these challenges with longitudinal data from the Alzheimer's disease neuroimaging initiative (ADNI) [9]. The number of subjects completing each planned longitudinal visit diminishes gradually as subjects drop out, and scan timings are highly variable. Recently, several methods have been proposed to appropriately examine this type of data in a (mass-)univariate fashion and using classical statistical techniques suitable for longitudinal designs, e.g., linear mixed effects (LME) models [10] and generalized estimating equations (GEE) [11].

To our knowledge, however, there exists no purpose-built machine learning method that would offer the ability to optimally handle serial data, particularly from real-life longitudinal designs. We note that the scenario we consider is different from time-series data, which also deals with temporal dynamics. Yet in time-series analysis (e.g., of financial data), which has received considerable attention in machine learning, e.g. [12], temporal processes are typically sampled densely and at uniform intervals. A common goal is to fully characterize a single process in order to make forecasts. Instead, the longitudinal scenario we consider here assumes that each subject has a separate temporal process, which has been sampled a small number of times, possibly at non-uniform intervals.

We adopt the framework of the recently proposed Relevance Voxel Machine (RVoxM) [5], which offers state-of-the-art image-based prediction for a range of clinical applications and has a publicly available implementation. RVoxM builds on the Relevance Vector Machine (RVM) [13], a sparse Bayesian learning technique, and adapts it to model the spatial smoothness in images. Like virtually all machine learning algorithms, both RVM and RVoxM treat each datapoint as an independent sample, likely making their use sub-optimal for longitudinal data analysis. Inspired by LME models [10], we propose to introduce subject-specific random effects into the RVoxM model in order to capture the within-subject correlation structure in longitudinal data. Section 2 introduces the theoretical concepts of the proposed method. Section 3 presents empirical results and Section 4 provides a conclusion and discussion.

2 Theory

We aim to predict a target variable, $t \in \mathbb{R}$, from an image scan, $\boldsymbol{x}$, which denotes vectorized voxel values. We append 1 to $\boldsymbol{x}$ to account for the bias. Thus $\boldsymbol{x}$ is $V+1$ dimensional, where V is the number of voxels. As in RVoxM [5], we assume:

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon, \tag{1}$$

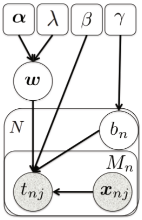

Fig. 1. Graphical model depicting the dependency structure between model variables. Circles represent random variables and model parameters are in squares. Plates indicate replication. Shaded variables are observed (during training), whereas remaining variables are unknown (latent). For variable names and further details, refer to text.

where the error is zero-mean Gaussian, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$, with variance $\beta^{-1} > 0$.

Similar to RVoxM we adopt a linear model for y. Unlike RVoxM, however, we utilize subject-specific random variables to account for the within-subject correlation structure in longitudinal data. Thus, we assume:

$$y_{nj} = \boldsymbol{x}_{nj}^{\mathrm{T}} \boldsymbol{w} + b_n, \tag{2}$$

where the subscripts n and j denote the subject and subject-specific time-point indices, respectively; $\boldsymbol{w}$ is the vector of latent model coefficients shared across subjects (one for each voxel and a bias term); and b_n is the latent, subject-specific bias term. As in RVoxM [5], the coefficients are assumed to be drawn from a sparsity-inducing, spatial-smoothness-encouraging prior:

$$\boldsymbol{w} \sim N(\mathbf{0}, \mathbf{P}^{-1}), \tag{3}$$

where $\mathbf{P} = \mathrm{diag}(\boldsymbol{\alpha}) + \lambda L$, $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_{V+1}] > 0$, and $\lambda \geq 0$ are hyperparameters; and L is a Laplacian matrix defined as:

$$L(u,v) = \begin{cases} -1 & \text{, if } u \text{ and } v \text{ are indices of neighboring voxels} \\ \text{Number of neighbors} & \text{, if } u = v \\ 0 & \text{, otherwise.} \end{cases} \tag{4}$$

The critical component of Eq. 2 is the subject-specific bias term b_n, which we assume to be drawn from a zero-mean Gaussian, $b_n \sim N(0, \gamma^{-1})$, with variance $\gamma^{-1} > 0$. Fig. 1 shows the graphical model illustrating the relationship between model variables. Similar to LME models, b_n captures the positive correlation between serial datapoints on the same individual. That is, we expect the error of the core prediction algorithm (e.g., RVoxM with $b_n = 0$) to be positively correlated for longitudinal data. So, for example, if the prediction is smaller than the ground truth (i.e., negative error) for the first time-point of a subject, we expect the error to be negative for a second time-point of the same individual too. The subject-specific bias term is intended to correct for this error. Empirical evidence presented in Fig. 2 supports our theoretical expectation of positive correlation between the prediction errors made on serial scans. The details of these data, which come from our empirical analysis, can be found in Section 3.

2.1 Training Phase

Let us assume we are given N training subjects, each with $M_n \geq 1$ time-points, where n denotes the subject index. Thus, we have $M = \sum_n M_n$ samples and

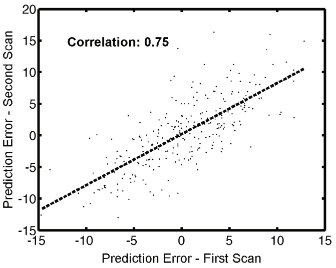

Fig. 2. Scatter plot of prediction error (predicted minus true value) of mini-mental state exam score (MMSE, a cognitive test that is associated with dementia). Prediction computed with RVoxM on brain MRI-derived cortical thickness data from ADNI. “First scan” was acquired at baseline and “Second scant” at the month 12 visit. Each point represents an individual. See experiment section for further details.

each sample consists of an image scan $\boldsymbol{x}$ and corresponding target variable value t. Let $\mathbf{X} \in \mathbb{R}^{M\times(V+1)}$ and $\boldsymbol{t} \in \mathbb{R}^{M}$ denote the stacked up training data. The goal of training is to learn the hyper-parameters $\boldsymbol{\alpha}$, λ, β and γ from the training data, e.g., via maximizing the type II likelihood:

$$\begin{aligned}(\boldsymbol{\alpha}^*, \lambda^*, \beta^*, \gamma^*) &= \underset{\boldsymbol{\alpha},\lambda,\beta,\gamma}{\operatorname{argmax}} p(\boldsymbol{t}|\mathbf{X}; \boldsymbol{\alpha}, \lambda, \beta, \gamma) \\ &= \underset{\boldsymbol{\alpha},\lambda,\beta,\gamma}{\operatorname{argmax}} \int p(\boldsymbol{w}; \boldsymbol{\alpha}, \lambda) \prod_n p(\boldsymbol{t}_n|\mathbf{X}_n, \boldsymbol{w}; \beta, \gamma) d\boldsymbol{w}, \end{aligned} \tag{5}$$

where $\boldsymbol{t}_n \in \mathbb{R}^{M_n}$ and $\mathbf{X}_n \in \mathbb{R}^{M_n\times(V+1)}$ are the stacked up data (target variables and images) for the n'th training subject. The likelihood term for the n'th subject can be derived by marginalizing over the unknown subject-specific bias:

$$p(\boldsymbol{t}_n|\mathbf{X}_n, \boldsymbol{w}; \beta, \gamma) = \int p(\boldsymbol{t}_n|\mathbf{X}_n, \boldsymbol{w}; \beta, b_n) p(b_n; \gamma) db_n = \mathcal{N}(\mathbf{X}_n^{\mathrm{T}} \boldsymbol{w}, \boldsymbol{\Lambda}_n), \tag{6}$$

where

$$\boldsymbol{\Lambda}_n = \left(\beta(\mathbf{I}_{M_n} - \frac{\beta}{\beta M_n + \gamma} \mathbf{1}_{M_n} \mathbf{1}_{M_n}^{\mathrm{T}})\right)^{-1} = \beta^{-1} \mathbf{I}_{M_n} + \gamma^{-1} \mathbf{1}_{M_n} \mathbf{1}_{M_n}^{\mathrm{T}}. \tag{7}$$

$\mathbf{I}_{M_n}$ denotes the $M_n \times M_n$ identity matrix and $\mathbf{1}_{M_n}$ denotes a length M_n column vector of 1's. Note $\boldsymbol{\Lambda}_n$ is in general not a diagonal matrix, thus modeling correlation structure between serial data. Inserting Eq. 6 into Eq. 5 and working out the integral yields $p(\boldsymbol{t}|\mathbf{X}; \boldsymbol{\alpha}, \lambda, \beta, \gamma) = \mathcal{N}(\mathbf{0}, \mathbf{C})$, where $\mathbf{C} = \boldsymbol{\Lambda} + \mathbf{X}^{\mathrm{T}} \mathbf{P}^{-1} \mathbf{X}$, $\boldsymbol{\Lambda} = \mathbf{I}_N \otimes \boldsymbol{\Lambda}_n$ and $\otimes$ is the Kronecker product. Note $\mathbf{C}$ depends on all hyper-parameters $(\boldsymbol{\alpha}, \lambda, \beta, \gamma)$ and the optimization (learning) problem of Eq. 5 can be re-written as:

$$\underset{\boldsymbol{\alpha},\lambda,\beta,\gamma}{\operatorname{argmin}} \boldsymbol{t}^{\mathrm{T}} \mathbf{C}^{-1} \boldsymbol{t} + \log |\mathbf{C}|, \tag{8}$$

where $|\cdot|$ denotes matrix determinant. Let's define:

$$\boldsymbol{\Sigma} = (\mathbf{P} + \mathbf{X}^{\mathrm{T}} \boldsymbol{\Lambda}^{-1} \mathbf{X})^{-1}, \text{ and } \boldsymbol{\mu} = \boldsymbol{\Sigma} \mathbf{X}^{\mathrm{T}} \boldsymbol{\Lambda}^{-1} \boldsymbol{t}. \tag{9}$$

Following the algebraic manipulations of [5,13], we arrive at the following update equation for the elements of the hyper-parameter vector $\boldsymbol{\alpha}$:

$$\alpha_i \leftarrow \frac{1 - \alpha_i \boldsymbol{\Sigma}_{ii} - \lambda \left(\mathbf{P}^{-1} \mathbf{L}\right)_{ii}}{\mu_i^2}, \tag{10}$$

which satisfies the non-negativity constraints and optimizes Eq. 8. As originally observed by Tipping [13], this optimization procedure tends to yield many α_i's that diverge to ∞ in practice, effectively turning off the contribution of the corresponding voxel (see next subsection) and producing a sparse model. We estimate the three scalar hyper-parameters, (β, λ, γ), by solving Eq. 8 via gradient-descent.

2.2 Testing Phase

The training phase provides a set of learned hyper-parameters $(\boldsymbol{\alpha}^*, \lambda^*, \beta^*, \gamma^*)$. The testing phase involves computing the posterior distribution of the target variable given longitudinal image data from a novel test subject, $\mathbf{X}_{N+1} \in \mathbb{R}^{M_{N+1}\times(V+1)}$, with $M_{N+1} \geq 1$ time-points. It can be shown that this posterior is a Gaussian:

$$p(\boldsymbol{t}_{N+1}|\mathbf{X}_{N+1}; \boldsymbol{\alpha}^*, \lambda^*, \beta^*, \gamma^*) = \mathcal{N}(\mathbf{X}_{N+1}\boldsymbol{\mu}^*, \tilde{\boldsymbol{\Sigma}}_{N+1}), \tag{11}$$

where

$$\tilde{\boldsymbol{\Sigma}}_{N+1} = \boldsymbol{\Lambda}^*_{N+1} + \mathbf{X}_{N+1}\boldsymbol{\Sigma}^*\mathbf{X}^{\mathrm{T}}_{N+1}, \tag{12}$$

$\boldsymbol{\Lambda}^*_{N+1} = (\beta^*)^{-1}\mathbf{I}_{M_{N+1}} + (\gamma^*)^{-1}\mathbf{1}_{M_{N+1}}\mathbf{1}^{\mathrm{T}}_{M_{N+1}}$, and $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$ are computed based on Eq. 9 with learned hyper-parameters. From Eq. 11, the maximum a posteriori probability (MAP) estimate of the target variable is:

$$\hat{\boldsymbol{t}}_{N+1} = \mathbf{X}_{N+1}\boldsymbol{\mu}^*. \tag{13}$$

Using Eq 9 it can be shown that if $\alpha_i \to \infty$, then $\mu_i = 0$. Thus, the corresponding voxel has no influence on the MAP prediction.

In certain scenarios with $M_{N+1} > 1$, target variable values might be available for some time-points of the test subject. Without loss of generality, let us decompose $\boldsymbol{t}_{N+1} = [\boldsymbol{t}^{\text{known}}_{N+1}; \boldsymbol{t}^{\text{unknown}}_{N+1}]$, where the superscript indicates whether the target variable is known or not. Using well-known formulae for conditional multivariate Gaussians, the MAP estimate for $\boldsymbol{t}^{\text{unknown}}_{N+1}$ can be written as:

$$\hat{\boldsymbol{t}}^{\text{unknown}}_{N+1} = \mathbf{X}^{\text{unknown}}_{N+1}\boldsymbol{\mu}^* + [\tilde{\boldsymbol{\Sigma}}^{\text{unknown, known}}_{N+1}][\tilde{\boldsymbol{\Sigma}}^{\text{known, known}}_{N+1}]^{-1}(\boldsymbol{t}^{\text{known}}_{N+1} - \mathbf{X}^{\text{known}}_{N+1}\boldsymbol{\mu}^*), \tag{14}$$

where we have used: $\tilde{\boldsymbol{\Sigma}}_{N+1} = \left[\begin{array}{c|c} \tilde{\boldsymbol{\Sigma}}^{\text{known, known}}_{N+1} & \tilde{\boldsymbol{\Sigma}}^{\text{known, unknown}}_{N+1} \\ \hline \tilde{\boldsymbol{\Sigma}}^{\text{unknown, known}}_{N+1} & \tilde{\boldsymbol{\Sigma}}^{\text{unknown, unknown}}_{N+1} \end{array}\right]$.

2.3 Implementation

The learning algorithm is initialized with: $\boldsymbol{\mu}^{\text{init}} = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}\boldsymbol{t}$, $\alpha^{\text{init}}_i = \frac{1}{(\mu^{\text{init}}_i)^2}$, $\lambda^{\text{init}} = 1$, $\beta^{\text{init}} = \frac{5}{\text{var}(\boldsymbol{t})}$, $\gamma^{\text{init}} = \frac{50}{\text{var}(\boldsymbol{t})}$. The objective function of Eq. 8 is monitored at each iteration and the optimization terminates once the change in the value is below a preset tolerance threshold. As RVoxM and RVM, the computational demands of the proposed algorithm are significant. A naive implementation, for example, can require $\mathcal{O}(V^3)$ time, and V, the number of voxels, can reach hundreds of thousands. Instead, we use a greedy algorithm, originally proposed for RVM [13], that permanently "turns off" voxels when their corresponding α_i exceeds a threshold (e.g., 10^{12}). Finally, all update equations can be expressed only using effective voxels (that haven't been turned off) and we can exploit the sparsity of $\mathbf{P}$ to speed up some matrix operations, as described in [5].

Our implementation is based on publicly available RVoxM code in Matlab. We call the proposed algorithm LRVoxM, for longitudinal RVoxM. We implemented a surface-based version designed to handle FreeSurfer-derived cortical surface data. FreeSurfer [14] is a freely available toolkit for automatically processing brain MRI scans. For example, given a structural brain MRI scan, FreeSurfer can compute a cortical thickness map sampled onto a common template surface mesh that represents a population average.

3 Empirical Results

In our experiment, we considered the problem of predicting the mini-mental state exam score (MMSE ranges between 0-30 and a lower score is associated with heightened dementia risk) from a 1.5T structural brain MRI scan. We analyzed longitudinal data from ADNI [9]. We generated two groups that consisted of pairs of matched subjects with three or more longitudinal scans at > 6 month intervals. The matching was done based on baseline characteristics (age, sex, and diagnosis), yielding two groups of $N = 273$ with the following composition: 77 ± 7.1 years, %40 female, 76 healthy controls, 135 subjects with mild cognitive impairment (MCI, a clinical stage associated with high dementia risk), and 62 Alzheimer's disease (AD) patients. Total number of scans across both groups was 3069, i.e., an average of 5.62 MRI scans per subject. We assigned an MMSE score to each scan based on the clinical assessment closest to the MRI date.

Each matched pair of subjects were randomly split into a test subject and a training subject. We repeated this random split 10 times to obtained 10 random matched train/test datasets. Note that, since each subject contained a variable number of (≥ 3) longitudinal time-points, the sizes of the train/test datasets were in general different.

The input to the prediction algorithm was FreeSurfer-derived cortical thickness data sampled on a common template surface (*fsaverage6* with > 70k vertices) and smoothed with a Gaussian of FWHM = 5mm. As the baseline method, we considered the public implementation of RVoxM, which corresponded to setting $\gamma = \infty$ in the proposed LRVoxM algorithm. For LRVoxM, we examined three test phase scenarios: (**S0**) target variable was unknown for all test subject scans (LRVoxM-0), (**S1**) target variable was given for first scan (LRVoxM-1), and (**S2**) first two scans (LRVoxM-2) of each test subject. For **S1** and **S2**, LRVoxM used Eq. 14 to compute predictions. In assessing testing accuracy, we only used the scans for which none of the algorithms had access to the target variable (i.e., we excluded first two scans of each subject). Note that testing accuracy metrics for LRVoxM-0 and RVoxM remained virtually unchanged when we included the first two scans of each subject.

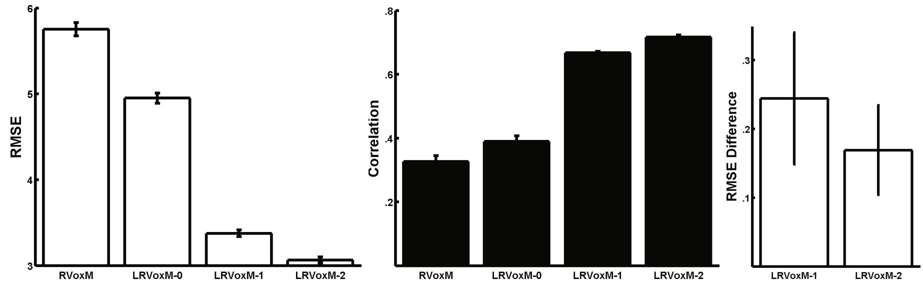

Fig. 3. Average testing accuracy (left: RMSE, middle: Correlation) in predicting MMSE from MRI scan. RVoxM: baseline method. LRVoxM: proposed longitudinal RVoxM algorithm. Right: Avg. RMSE improvement over MRI-blind predictions in scenarios **S1** and **S2** (see text). The bar charts show the mean values over the 10 test phase sessions. Errorbars indicate standard error of the mean.

Fig. 2 shows a scatter plot of the prediction error (predicted minus true MMSE value) of RVoxM, the benchmark algorithm, on the baseline and year 1 scans of a test dataset. This plot reveals the strong positive correlation of the prediction error in longitudinal data, which was the core motivation of our approach. The main empirical results are presented in Fig. 3, which shows the root mean squared error (RMSE) and correlation between the predictions and ground truth values. LRVoxM-0 clearly outperforms RVoxM on both metrics (avg. RMSE: 4.95 v. 5.75, avg. correlation: 0.39 v. 0.33, paired t-test across 10 sessions, $p < 10^{-7}$), which demonstrates the accuracy boost achieved by appropriately accounting for longitudinal data in a prediction algorithm.

Moreover, LRVoxM-1 and LRVoxM-2 offer progressively better prediction performance. For example, providing the target variable value for only a single time-point reduces RMSE by about 32% to an average of 3.38, whereas LRVoxM-2 achieves a further improved average RMSE of 3.07. These results demonstrate that LRVoxM exploits subject-specific information to compute improved predictions (via the use of Eq. 14). We conducted an additional comparison of the LRVoxM results with "MRI-blind" predictions, which were computed based on the available MMSE values for each test subject (scenarios **S1** and **S2**, see Fig. 3-right). LRVoxM-1's RMSE was significantly lower than a prediction that assigned the first time point's MMSE value to all remaining time-points of the same subject (paired t-test $p < 0.02$). For **S2**, the MRI-blind prediction was computed by fitting a line to the given MMSE values of the first two time-points. The predictions were then computed from this line at subsequent visits. LRVoxM-2's RMSE was also significantly lower than this MRI-blind prediction ($p < 0.04$).

4 Discussion and Conclusion

We presented a novel, sparse Bayesian learning algorithm suitable for longitudinal image data. Our experiments demonstrated a significant boost in performance achieved by the proposed method in comparison with the conventional strategy that ignores the longitudinal structure. Although we utilized the RVoxM framework, our approach is general and can be adopted within alternative probabilistic models, e.g., Gaussian processes, or for other data types. Future work will include extending LRVoxM to handle discrete target variables (classification), compute predictions about future outcome (prognosis), and examine alternative priors that might be more appropriate for handling discrete jumps in spatial data. We further plan to explore the use of additional subject-specific terms (such as those depending on time) to capture more complex correlation patterns, e.g, that depend on the time interval between visits.

References

1. Davatzikos, C., et al.: Individual patient diagnosis of AD and FTD via high-dimensional pattern classification of MRI. Neuroimage 41(4), 1220–1227 (2008)
2. Gaonkar, B., Davatzikos, C.: Deriving statistical significance maps for svm based image classification and group comparisons. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 723–730. Springer, Heidelberg (2012)
3. Sabuncu, M.R., et al.: Clinical prediction from structural brain MRI scans: A large-scale empirical study. Neuroinformatics, 1–16 (2014)
4. Cuingnet, R., et al.: Spatial regularization of SVM for the detection of diffusion alterations associated with stroke outcome. Medical Image Analysis, 15 (2011)
5. Sabuncu, M.R., Van Leemput, K.: The Relevance Voxel Machine (RVoxM): A self-tuning bayesian model for informative image-based prediction. IEEE Transactions on Medical Imaging (2012)
6. Suk, H.-I., Shen, D.: Deep learning-based feature representation for AD/MCI classification. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013, Part II. LNCS, vol. 8150, pp. 583–590. Springer, Heidelberg (2013)
7. Liu, M., et al.: Identifying informative imaging biomarkers via tree structured sparse learning for AD diagnosis. Neuroinformatics 12(3) (2014)
8. Jie, B., et al.: Manifold regularized multitask feature learning for multimodality disease classification. Human Brain Mapping 36(2) (2015)
9. Jack, C.R., et al.: The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods. Journal of Magnetic Resonance Imaging 27(4) (2008)
10. Bernal-Rusiel, J., et al.: Statistical analysis of longitudinal neuroimage data with linear mixed effects models. Neuroimage 66 (2013)
11. Li, Y., et al.: Multiscale adaptive generalized estimating equations for longitudinal neuroimaging data. Neuroimage 72 (2013)
12. Cao, L.-J., Tay, F.E.H.: Support vector machine with adaptive parameters in financial time series forecasting. IEEE T. on Neural Networks (2003)
13. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211–244 (2001)
14. Fischl, B.: FreeSurfer. Neuroimage 62(2), 774–781 (2012)