An Improved Optimization Method for the Relevance Voxel Machine

Melanie Ganz[1,2], Mert R. Sabuncu[1], and Koen Van Leemput[1,3,4]

[1] Martinos Center for Biomedical Imaging, MGH, Harvard Medical School, USA
[2] Department for Computer Science, University of Copenhagen, Denmark
[3] Department of Applied Mathematics and Computer Science, DTU, Denmark
[4] Departments of Information and Computer Science and of Biomedical Engineering and Computational Science, Aalto University, Finland

Abstract. In this paper, we will re-visit the Relevance Voxel Machine (RVoxM), a recently developed sparse Bayesian framework used for predicting biological markers, e.g., presence of disease, from high-dimensional image data, e.g., brain MRI volumes. The proposed improvement, called IRVoxM, mitigates the shortcomings of the greedy optimization scheme of the original RVoxM algorithm by exploiting the form of the marginal likelihood function. In addition, it allows voxels to be added and deleted from the model during the optimization. In our experiments we show that IRVoxM outperforms RVoxM on synthetic data, achieving a better training cost and test root mean square error while yielding sparser models. We further evaluated IRVoxM's performance on real brain MRI scans from the OASIS data set, and observed the same behavior - IRVoxM retains good prediction performance while yielding much sparser models than RVoxM.

1 Introduction

Multivariate pattern analysis (MVPA) methods provide an alternative approach to examining subtle and complex relationships between biomedical images, e.g., structural [1] or functional [2] neuroimage data, and clinical variables of interest. They further offer the ability to make accurate individual-level predictions of clinical outcome, paving the way to personalized medicine. One specific class of MVPA algorithms employs sparse Bayesian learning; a well-known algorithm of that class is the Relevance Vector Machine (RVM) [3] which is similar to a support vector machine (SVM) [4]. In contrast to SVM, it provides probabilistic outcomes and has no free parameters to tune. Problems with RVM are the computational time, which is approximately cubic in the number of inputs, as well as the greedy optimization. Thus different approaches have been developed to overcome these shortcomings, e.g. FastRVM [5].
Recently, Sabuncu and Van Leemput [6, 7] extended the relevance vector machine by incorporating an additional spatial regularization term in the Gaussian prior on the regression weights or classification features (RVoxM). RVoxM encourages spatial clustering of the features used for regression or classification, and computes predictions as linear combinations of their content. While the model of RVoxM produced nice results on age regression data [6, 7], the algorithm used a simple fixed point optimization scheme,

which is not guaranteed to decrease the cost function at every step and is computationally expensive. In addition, RVoxM prunes voxels from the linear model by applying an artificial numerical threshold to the weight hyperparameters, which creates a free parameter that affects model sparsity. Lastly, RVoxM can only remove voxels from the model, but not re-introduce them later on. Thus in its current form, it is reminiscent of a greedy forward feature selection algorithm.
In this paper, we aim to solve the problems of the original RVoxM algorithm in the spirit of FastRVM [5]. We call the new algorithm Improved Relevance Voxel Machine (IRVoxM). Our contributions improve the greedy optimization algorithm of RVoxM by deriving an analytic expression for the optimal hyperparameter of each voxel, given the current hyperparameter of all other voxels. This enables us to maximize the marginal likelihood function in a principled and efficient manner. As a result, IRVoxM optimizes the objective function better during training, and the resulting models predict better on unseen cases. Lastly, IRVoxM enables us to easily add and/or remove voxels during the optimization procedure, allowing us to start from an empty model which has the potential for large computational advantages.

2 Regression with the Relevance Voxel Machine - RVoxM

We base IRVoxM on the same theoretical model as RVoxM [6, 7]. In the regression problem, the target variable t, e.g., age or clinical test score, is assumed to be Gaussian distributed:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}), \tag{1}$$

with variance β^{-1} and mean $y(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{M-1} x_i w_i + w_M = \mathbf{w}^{\mathrm{T}}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^M$ is a vector that represents the input data, e.g., an image, plus a constant element of one ($x_M = 1$), and $\mathbf{w} \in \mathbb{R}^M$ are weights.
We further assume a Gaussian prior on $\mathbf{w}$ with hyperparamters $\boldsymbol{\alpha}$ and λ of the form

$$p(\mathbf{w}|\boldsymbol{\alpha}, \lambda) = \mathcal{N}(\mathbf{w}|0, \mathbf{P}^{-1}), \tag{2}$$

where $\mathbf{P} = \mathrm{diag}(\boldsymbol{\alpha}) + \lambda\mathbf{K}$. $\mathbf{K} = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma}$ is the graph Laplacian matrix which is a sparse, symmetric matrix and can be defined as the inner product of the incidence matrix $\boldsymbol{\Gamma}$. $\boldsymbol{\Gamma}$ is a sparse matrix of dimension $N_{\mathrm{Edg}} \times M$, where N_{Edg} denotes the number of edges in the graph spanned by $\mathbf{K}$. Each row of $\boldsymbol{\Gamma}$ has only two entries that denote the outgoing $(+1)$ and incoming (-1) nodes of an edge in the graph. In our case, edges connect physically neighboring locations, e.g., all voxels in the 6-neighborhood are connected to a central voxel in a volumetric image, but other configurations can also be considered. $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^{\mathrm{T}}$ and λ are hyperparameters; the α_i are inverse covariances of the weight prior and hence control the sparsity of the weights. A large α_i means the weight w_i of the associated voxel is tending to zero, while a small α_i implies that the value w_i is largely determined by its neighbors. The parameter λ encourages spatial smoothness and the larger it is the smoother the resulting weight maps are.

Training With the above prior, the hyperparameters can be estimated by maximizing the following type-II likelihood given a collection of training target values $\mathbf{t} =$

$(t_1, \ldots, t_N)^{\mathrm{T}}$ and a set of N training images $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}}$:

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}, \boldsymbol{\alpha}, \beta, \lambda) &= \int_{\mathbf{w}} p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w}|\boldsymbol{\alpha}, \lambda) d\mathbf{w} \\ &= \int_{\mathbf{w}} \left(\prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta) \right) p(\mathbf{w}|\boldsymbol{\alpha}, \lambda) d\mathbf{w} = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}), \end{aligned} \tag{3}$$

where we define $\mathbf{C} = \beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{P}^{-1}\mathbf{X}^{\mathrm{T}}$. We can estimate the hyperparameters $\boldsymbol{\alpha}, \beta, \lambda$, which is equivalent to maximizing Eq. 4:

$$\hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda} = \underset{\boldsymbol{\alpha}, \beta, \lambda}{\operatorname{argmax}} \, \mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda) = \underset{\boldsymbol{\alpha}, \beta, \lambda}{\operatorname{argmax}} \left(-\frac{1}{2} (N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{t}) \right). \tag{4}$$

Here, $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ denotes the logarithm of the marginal likelihood function, which is obtained by integrating out the weight parameters as shown in Eq. 3. In RVoxM [6], this optimization was solved by a coordinate ascent over β and λ, while optimizing over all $\boldsymbol{\alpha}$ simultaneously using a fixed point equation and a greedy approach, where single α_i's exceeding a numerical threshold are pruned from the model. This optimization of $\boldsymbol{\alpha}$ has no theoretical guarantees of convergence and is computationally expensive. Hence, in this paper we focus on deriving a better optimization algorithm for $\boldsymbol{\alpha}$.

Prediction After obtaining $\hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda}$ from training data, we can make predictions for a new $\mathbf{x}^\star$ according to

$$p(t^\star|\mathbf{x}^\star, \mathbf{X}, \mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda}) = \int p(t^\star|\mathbf{x}^\star, \mathbf{w}, \hat{\beta}) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\lambda}) d\mathbf{w} = \mathcal{N}(\mu^\star, \Sigma^\star), \tag{5}$$

where $p(t^\star|\mathbf{x}^\star, \mathbf{w}, \hat{\beta})$ is given by the regression model in Eq. 1 and $\mu^\star = \boldsymbol{\mu}^T \mathbf{x}$ and $\Sigma^\star = \frac{1}{\beta} + \mathbf{x}^{\mathrm{T}} \boldsymbol{\Sigma} \mathbf{x}$, in which $\boldsymbol{\Sigma} = (\mathbf{P} + \beta \mathbf{X}^{\mathrm{T}} \mathbf{X})^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma} \mathbf{X}^{\mathrm{T}} \mathbf{t}$.

3 The Improved Relevance Voxel Machine - IRVoxM

The greedy optimization method employed by RVoxM suffers from several weaknesses. First of all, the fixed point updates used for $\boldsymbol{\alpha}$ in [6] are not guaranteed in each update step to increase the logarithm of the marginal likelihood $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ given in Eq. 4. Thus the optimization can become unstable, if it is not well initialized. Another difficulty is the use of an artificial numerical threshold that controls the sparsity of the solution by deciding at which point an α_i is considered to be large enough to yield a weight w_i close enough to zero to be effectively pruned from the model. Finally, RVoxM works by pruning voxels from a larger set until only relevant voxels are left. But once voxels are removed from the model there is no way to re-introduce them; the set of voxels included in the sparse model can only decline.

In this paper, we present a way of tackling all of the above issues. First, we ensure that single updates are guaranteed to increase the logarithm of the marginal likelihood. Then we introduce a conceptual change that allows us to analytically judge if a voxel

should be included in the model or not. This change also enables us to re-visit and re-introduce voxels into our sparse model during the optimization, which addresses the third problem. To derive the improved Relevance Voxel Machine (IRVoxM) we study $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ for fixed β and λ; thus $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ is only dependent on $\boldsymbol{\alpha}$. We can rewrite the logarithm of the marginal likelihood function $\mathcal{L}(\boldsymbol{\alpha})$ (Eq. 4) to expose the marginal contribution of α_i.
But first, let us introduce some notation:

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{\Gamma} \end{pmatrix}, \quad \tilde{\mathbf{t}} = \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix}, \quad \tilde{\mathbf{B}} = \begin{pmatrix} \beta \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \lambda \mathbf{I}_M \end{pmatrix}. \tag{6}$$

Now we can define $\tilde{\mathbf{C}} = \tilde{\mathbf{B}}^{-1} + \tilde{\mathbf{X}} \mathrm{diag}(\boldsymbol{\alpha}^{-1}) \tilde{\mathbf{X}}^{\mathrm{T}}$ and therefore $|\tilde{\mathbf{C}}| = \beta^{-N} \lambda^{-N_{\mathrm{Edg}}} \frac{|\mathbf{\Sigma}|}{|\mathrm{diag}(\boldsymbol{\alpha})|}$.
In addition, we write $\tilde{\mathbf{C}}^{-1} = \tilde{\mathbf{B}} - \tilde{\mathbf{B}} \tilde{\mathbf{X}} \tilde{\mathbf{\Sigma}} \tilde{\mathbf{X}}^{\mathrm{T}} \tilde{\mathbf{B}}$ and express the logarithm of the marginal likelihood (Eq. 4) as:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\alpha}) &= -\frac{1}{2} \left(N \ln(2\pi) + \ln(\beta^{-N} \frac{|\mathbf{\Sigma}|}{|\mathbf{P}|}) + \mathbf{t}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{t} \right) \\
&= -\frac{1}{2} \left(N \ln(2\pi) + \ln|\tilde{\mathbf{C}}| + \tilde{\mathbf{t}}^{\mathrm{T}} \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{t}} \right) - \frac{1}{2} \left(N_{\mathrm{Edg}} \ln(\lambda) - \ln|\mathbf{P}| + \ln|\mathrm{diag}(\boldsymbol{\alpha})| \right) \\
&= \underbrace{-\frac{1}{2} \left(N \ln(2\pi) + \ln|\tilde{\mathbf{C}}_{-i}| + \tilde{\mathbf{t}}^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{t}} + N_{\mathrm{Edg}} \ln(\lambda) - \ln(|\mathbf{\Psi_{-i}}|) \right)}_{\mathcal{L}(\boldsymbol{\alpha}_{-i})} \\
&\quad \underbrace{+\frac{1}{2} \left(-\ln(\alpha_i + \tilde{s}_i) + \frac{\tilde{q}^2}{\alpha_i + \tilde{s}_i} + \ln(\alpha_i + a_i) \right)}_{l(\alpha_i)}.
\end{aligned} \tag{7}$$

As in [5], we first use the definitions of $|\mathbf{C}|$, $|\tilde{\mathbf{C}}|$ and $\tilde{\mathbf{C}}^{-1}$ to rewrite Eq. 4 and then employ established matrix determinant and inverse identities to separate the contribution of α_i from $\tilde{\mathbf{C}}$ and its inverse. This yields $\tilde{\mathbf{C}}_{-i}$ and $\tilde{\mathbf{C}}_{-i}^{-1}$ which are $\tilde{\mathbf{C}}$ and its inverse with the contribution of the i-th voxel removed. Lastly, we define

$$\tilde{s}_i = \tilde{\mathbf{X}}_i^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{X}}_i, \tilde{q}_i = \tilde{\mathbf{X}}_i^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{t}}, \mathbf{\Psi_{-i}} = \mathbf{I} + \sum_{j \neq i} \frac{\lambda}{\alpha_j} \mathbf{\Gamma}_j \mathbf{\Gamma}_j^{\mathrm{T}}, a_i = \frac{\lambda \mathbf{\Gamma}_i \mathbf{\Psi_{-i}}^{-1} \mathbf{\Gamma}_i^{\mathrm{T}}}{\alpha_i}, \tag{8}$$

in which $\tilde{\mathbf{X}}_i$ and $\mathbf{\Gamma}_i$ denote the i-th column of $\tilde{\mathbf{X}}$ and $\mathbf{\Gamma}$, respectively.
In Eq.7 we have replaced $\mathcal{L}(\boldsymbol{\alpha})$ by $\mathcal{L}(\boldsymbol{\alpha}_{-i})$ and $l(\alpha_i)$. $\mathcal{L}(\boldsymbol{\alpha}_{-i})$ includes the contributions of all $\boldsymbol{\alpha}$ except for the i-th α to the marginal likelihood, whereas $l(\alpha_i)$ comprises only the contribution of α_i. Equation 7 can now be optimized with respect to α_i in closed form (while keeping all other $\boldsymbol{\alpha}$'s fixed to their current values):

1. If $a_i \geq \tilde{s}_i$, the optimal solution is $\hat{\alpha}_i = 0$, since we do not allow negative alphas as in [6, 7].
2. If $a_i < \tilde{s}_i$ and $\tilde{s}_i - a_i < \tilde{q}_i^2$, the optimal solution is $\hat{\alpha}_i = \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}$. If this solution is negative, the optimal solution is $\hat{\alpha}_i = 0$.
3. If $a_i < \tilde{s}_i$ and $\tilde{s}_i - a_i \geq \tilde{q}_i^2$, the optimal solution is $\hat{\alpha}_i = \infty$, which corresponds to removing the i'th voxel from the model (see [3, 6, 7]).

Algorithm 1 IRVoxM algorithm

Initialize λ, β and all $\boldsymbol{\alpha}$ as in RVoxM [7].
loop
  loop
    Randomly pick a voxel i.
    Compute $\tilde{s}_i$,$\tilde{q}_i$ and a_i according to Eqs. 8.
    if $a_i \geq \tilde{s}_i$ **then**
      $\alpha_i = 0$
    else if $a_i < \tilde{s}_i$ **then**
      if $\tilde{s}_i - a_i < \tilde{q}_i^2$ **then**
        $\alpha_i = \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}$
        if $\alpha_i < 0$ **then**
          $\alpha_i = 0.$
        end if
      else if $\tilde{s}_i - a_i \geq \tilde{q}_i^2$ **then**
        $\alpha_i = \infty$
      end if
    end if
    Update all quantities in an efficient manner as derived in [8].
  end loop
  Update β and λ by a simple search of the two-dimensional cost function.
end loop

The above rules enable us to loop over all voxels and update a single α_i at a time.
An overview of the IRVoxM algorithm is given in Algorithm 1. Details regarding the derivation of $l(\alpha_i)$ as well as a functional analysis of the marginal likelihood function are excluded due to space constraints, but can be found in a technical report [8].

4 Experiments and Results

In order to demonstrate that our proposed optimizer outperforms RVoxM's, we will evaluate the performance of IRVoxM and RVoxM on a synthetic and a real data set. To make the comparison fair, we initialize the two algorithms identically with $\boldsymbol{\alpha} = \mathbf{1}$, $\beta = 1$ and $\lambda = 1$.

4.1 Experiments on synthetic data

First, we ran experiments on synthetic data. To model a single target value t, we generated a random vectorized image $\mathbf{x}$ by drawing random samples from a Gaussian distribution with mean 0 and standard deviation 1 of size $M \times 1$. Using pre-determined constants $\boldsymbol{\alpha}_{\text{true}} = (10^{12}\mathbf{v}, 0.5\mathbf{v}, 10^{12}\mathbf{v})^{\text{T}}$, where $\mathbf{v}$ is a vector of ones and of dimension $\frac{M}{3} \times 1$, and $\lambda_{\text{true}} = 10$, we constructed $\mathbf{P}_{\text{true}} = \text{diag}(\boldsymbol{\alpha}_{\text{true}}) + \lambda_{\text{true}}\boldsymbol{\Gamma}^{\text{T}}\boldsymbol{\Gamma}$. Here, $\boldsymbol{\Gamma}$ is the incidence matrix for a 4-neighborhood. From $\mathbf{P}_{\text{true}}$ we sampled weights $\mathbf{w}_{\text{true}}$ and computed targets as $t = \mathbf{w}_{\text{true}}^{\text{T}}\mathbf{x} + \epsilon$, where the noise ϵ was sampled from a normal distribution with mean zero and inverse variance $\beta_{\text{true}} = 10$. We constructed data this

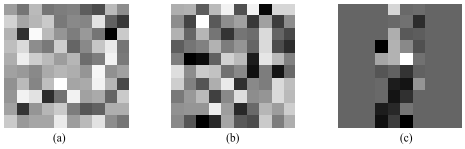

Fig. 1. Examples of two random images (a) and (b) as well as the weight vector (c) we used in our synthetic data experiment.

way for a varying number of training images N, yielding collections of image vectors $\mathbf{X}$ of size $N \times M$ as well as vectors of target values $\mathbf{t}$ of size $N \times 1$. We used an image size $M = 10 \times 10$. Lastly, we varied N from 10 to 100 and generated 100 independent pairs of $\mathbf{X}$ and $\mathbf{t}$ with the same weight vector $\mathbf{w}_{\mathrm{true}}$ for each value of N. For the test data, we generated another 100 independent pairs of $\mathbf{X}$ and $\mathbf{t}$ using $N = 100$, and applied the same weight vector $\mathbf{w}_{\mathrm{true}}$ as for the training data. Examples of two random images and the weight vector we used can be seen in Fig. 1. Fig. 2 shows the sparsity of the trained models, the training cost, which is the negative logarithm of the marginal likelihood given in Eq. 4, and the root mean square error (RMSE) between the true and the predicted target values computed on the test data sets. It also shows a comparison of the predicted and true weights by showing the l_2-norm of the difference between the true and the predicted weights of the two algorithms.
The results reveal several weaknesses of the original RVoxM. First, while the true sparsity of our synthetic data is always 33% (since we set 1/3 of the 100 weights to be different from zero), RVoxM grossly overestimates the number of weights that are included in the model (see Figure 1 a). IRVoxM on the other hand produces sparser models, while still achieving a better training cost on the training data (see Figure 1 b). Hence IRVoxM is not over fitting to the training data, but finding sparse models that represent the data well. Furthermore, RVoxM and IRVoxM yield comparable RMSE on the test data with IRVoxM considerably outperforming RVoxM for larger N (see Figure 1 c). Finally, IRVoxM produces weights that are much closer to the true weights for all values of N (see Figure 1d).

4.2 Experiments on OASIS

Next, we ran experiments on the publicly available cross-sectional Open Access Series of Imaging Studies (OASIS) [9]. We used structural brain MRI scans (T1-weighted) from 336 healthy subjects and processed them with SPM8[5] to obtain spatially aligned gray matter maps. The gray matter density values were used as voxel-level input data $\mathbf{x}_i$. To lighten the computational burden for our experiments, we generated masks with $N = 1000, 5000, 10000$ voxels for subsequent analysis. To acquire these masks, we calculated a univariate correlation of each voxel with age (only on the training data set) and sorted the voxels based on their correlation value, as was done in [10]. Then we used only the top N voxels in the analysis. We randomly chose half of the subjects and trained a model for age regression on this training dataset. Then we used the trained

[5] http://www.fil.ion.ucl.ac.uk/spm/software/spm8/

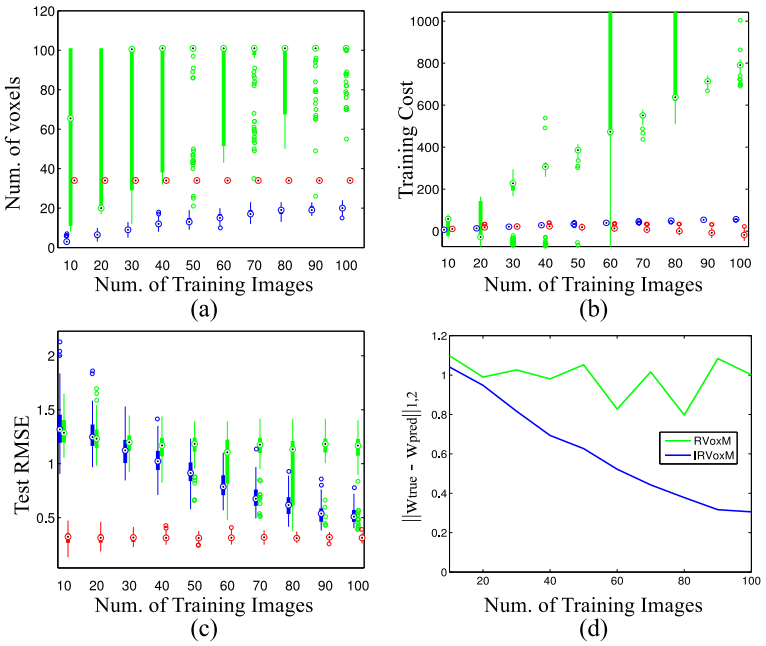

Fig. 2. Results for the synthetic data showing the resulting training sparsity (a), the training cost (b) and the root mean square error (RMSE) on the test data (c) for 100 independent repetitions. The box plots in (a), (b) and (c) show the ground truth (red), RVoxM (green) and IRVoxM (blue).Filled black dots indicate the median, filled boxes extend to the most extreme values within 1.5 times the interquartile range of the box. Lines extend to the adjacent value. Samples beyond those points are marked with colored circles.In (d) we show the l_2-norm of the differences between the true weights and the weights RVoxM produces (green) and the true weights and the weights IRVoxM produces (blue).

model to predict the age on the other half of the OASIS data set and calculated the correlation and the root mean square error (RMSE) between the predicted and the real age. The correlations and RMSE are shown in table 4.2. Again, we can observe that IRVoxM yields much sparser models than RVoxM, while achieving a high age correlation and low RMSE.

5 Discussion and Conclusion

We have re-visited the relevance voxel machine and introduced a better optimization scheme. By exploiting the form of the marginal likelihood function, we improved the way in which voxels are added and deleted from the sparse model during the optimization. Our algorithm IRVoxM outperforms RVoxM on synthetic data; it yields sparser models with good prediction performance and retains weight maps that are closer to the true synthetic weights than RVoxM's. On the OASIS data we have observed a similar

OASIS test set	RVoxM			IRVoxM		
168 images	1000	5000	10000	1000	5000	10000
Num. voxels in the model	173	176	151	42	44	17
Correlation	0.90	0.89	0.89	0.91	0.92	0.89
RMSE	10.47	10.85	10.82	10.17	9.97	11.02

Table 1. Model sparsity, test correlation and root mean square error (RMSE) for the OASIS data set including a different number of voxels in the mask.

behavior; while retaining good prediction performance, IRVoxM yields much sparser models than RVoxM.

Our aim in this paper was to show that our proposed algorithm IRVoxM improves over RVoxM's optimization scheme; thus we compared the two algorithms side by side. Our new optimization strategy performs as anticipated, and opens up a whole new avenue for speeding up computations, as was done previously for RVM [3] by FastRVM [5]. One key problem of RVoxM is the computational burden, especially during the first few iterations, where computational time is cubic in the number of voxels. IRVoxM does not need to be initialized with all voxels (as has been done for comparison to RVoxM in all our experiments here). One can start with only a few voxels in the model, which reduces the computational cost tremendously and preliminary experiments show that this approach performs equally well. Furthermore, our explicit functional formulation of the marginal likelihood function for a single α_i makes it possible to sample from the hyperparameter distributions, which had not been possible with RVoxM.

In further versions of IRVoxM, we plan to implement a different initialization strategy that enables us to increase the speed of IRVoxM, as well as exploit the possibility of sampling from the hyperparameter distribution.

6 Acknowledgments

This research was supported by the Alfred Benzon and the Lundbeck Foundation, NIH NCRR (P41-RR14075), NIBIB (R01EB013565), Academy of Finland (133611), TEKES (ComBrain), and the Technical University of Denmark.

References

1. Duarte, J.V., Ribeiro, M.J., Violante, I.R., Cunha, G., Silva, E., Castelo-Branco, M.: Multivariate pattern analysis reveals subtle brain anomalies relevant to the cognitive phenotype in neurofibromatosis type 1. Human Brain Mapping (2012)
2. Yang, Z., Fang, F., Weng, X.: Recent developments in multivariate pattern analysis for functional mri. Neuroscience bulletin **28**(4) (2012) 399–408
3. Tipping, M.: Sparse bayesian learning and the relevance vector machine. The Journal of Machine Learning Research **1** (2001) 211–244
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3) (1995) 273–297
5. Tipping, M.E., Faul, A.: Fast marginal likelihood maximisation for sparse bayesian models. In: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics. (2003) 3–6

6. Sabuncu, M., Leemput, K.: The relevance voxel machine (rvoxm): A bayesian method for image-based prediction. In Fichtinger, G., Martel, A., Peters, T., eds.: Medical Image Computing and Computer-Assisted Intervention MICCAI 2011. Volume 6893 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2011) 99–106
7. Sabuncu, M., Van Leemput, K.: The relevance voxel machine (rvoxm): A self-tuning bayesian model for informative image-based prediction. Medical Imaging, IEEE Transactions on **31**(12) (2012) 2290–2306
8. Ganz, M., Sabuncu, M.R., van Leemput, K.: The improved relevance voxel machine. Technical Report DTU Compute-2013-10, Institute for Mathematical Modeling, DTU (2013)
9. Marcus, D., Wang, T., Parker, J., Csernansky, J., Morris, J., Buckner, R.: Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. Journal of Cognitive Neuroscience **19**(9) (2007) 1498–1507
10. Michel, V., Eger, E., Keribin, C., Thirion, B.: Multiclass sparse bayesian regression for fmri-based prediction. Journal of Biomedical Imaging **2011** (2011) 2