Voxel-wise nonlinear analysis toolbox for neurodegenerative diseases and aging

Authors
Signal Theory and Communications Department
Universitat Politècnica de Catalunya, BarcelonaTech
Barcelona, Spain
`email`

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

2 The toolbox

The toolbox comprises an independent *fitting library*, made up of different *model fitting* and *fit evaluation* methods, a *processing* module that interacts with the aforementioned *fitting library* providing the formatted data obtained from the *file system*, several *visualization* tools and a *CLI interface* that allows the interaction between the user and the *processing* module, supported by a *configuration file*.

2.1 Model fitting techniques

A model fitting consists on finding a parametric or a nonparametric function of some explanatory variables (**predictors**) and possibly some confound variables (**correctors**) that best fits the observations of the target variable in terms of a given quality metric or, conversely, that minimizes the loss between the prediction of the model and the actual observations.

- **General Linear Model (GLM)**
  The General Linear model is a generalization of multiple linear regression to the case of more than one dependent variable. As in the case of multiple linear regression, the most common lost function is the Residual Sum of Squares, and the optimization procedure used is Ordinary Least Squares, which yields the well-known normal equation $X^T X\beta = X^T y$, and from that we solve for the β parameters to obtain the final solution: $\beta = (X^T X)^{-1} X^T y$.

  A possible approach to model nonliniearities with this model is a polynomial basis expansion of degree d, that is, the input space $\mathfrak{X}$ is mapped into another feature space $\mathfrak{F}$ that also includes the polynomial terms of the variables: ($\Phi : \mathfrak{X} \to \mathfrak{F}$).
- **Generalized Additive Model (GAM)**
  A Generalized Additive Model is a Generalized Linear Model in which the observations of the target variable depend linearly on unknown smooth functions of some predictor variables: $f(X) = \alpha + \sum_{i=1}^{k} f_i(X_i)$. Here $f_1, f_2, ..., f_k$ are nonparametric smooth functions that are simultaneously estimated using scatterplot smoothers by means of the **backfitting**

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

algorithm. Several fitting methods can be accomodated in this framework by using different smoother operators, such as cubic splines, polynomial or Gaussian smoothers.

- **Support Vector Regression (SVR)**

  The regression counterpart of the well-known Support Vector Machines, Support Vector Regression, is based on the following idea: the goal is to find a function that has at most ϵ deviation from the observations and, at the same time, is as flat as possible. However, the ϵ deviation contraint is not feasible sometimes, and a hyperparameter that controls the degree up to which deviations larger than ϵ are tolerated is introduced, C. The linear function for SVR is $f(x) = \langle w, x\rangle + b$, and then the solution for the optimization problem is $f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b$, where $(\alpha_i - \alpha_i^*)$ are the coefficients found in the dual optimization problem.

  In context of SVR the nonlinearities are introduced with the "kernel trick", that is, a kernel function $k(x_i, x_j) = \langle\Phi(x_i), \Phi(x_j)\rangle$ is introduced that implicitly maps the inputs from their original space into another high-dimensional space without requiring to know the explicit mapping $\Phi(\cdot)$. The solution using the kernel function is then $f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)k(x_i, x) + b$.

  The kernel function used in this toolbox is the Radial Basis Function or Gaussian kernel, which is defined as $k(x_i, x_j) = exp(-\gamma\|x_i - x_j\|^2)$.

2.2 Hyperparameters search algorithm

Support Vector Regression methods rely on several hyperparameters, namely ϵ and C in general and also γ when using a RBF kernel function. To address the search of these hyperparameters an automatic method based on grid search is included in this toolbox, which comprises the following steps: 1) sample the hyperparameters space in a grid using one of the several sampling methods provided in the toolbox; 2) fit a subset of the data with the combination of hyperparameters of each sample in the grid; 3) select the combination that minimizes the error function of choice.

The lack of validation data and the nature of morphometric data, which is vastly dominated by voxels with 0-valued observations, limits the ability to find a subset of the data that is valid to find the optimal hyperparameters without incurrying in overfitting or underfitting. For that reason the following approach is taken: a subset of m voxels is selected in each of the N iterations of the algorithm, the m selected voxels must contain observations with a minimum variance of Var_{min}, and the error computed for each voxel is weighted by the inverse of the variance of its observations.

2.3 Fit evaluation methods

- **F-test**

  The F-test can be used in regression problems to determine whether a particular part of a model is significantly improving the overall performance of the rest of the model. The F-test compares the variance of the error of the restricted model $M_{restricted}$, which consists only in the correctors, with the variance of the error of the full model M_{full}, which consists both in the correctors and the predictors, and evaluates whether the variance of the full model is significantly lower — from a statistical point of view — than the variance of the restricted model, that is, the inclusion of the predictors contributes to the explanation of the observations.

  The F-statistic is defined as:

$$F = \frac{\left(\frac{RSS_{restricted} - RSS_{full}}{df_{restricted} - df_{full}}\right)}{\left(\frac{RSS_{full}}{df_{full}}\right)} \tag{1}$$

  Under the null hypothesis F will follow a F-distribution of parameters $(df_{restricted} - df_{full}, df_{full})$.

  One can notice that this statistical test requires then the degrees of freedom of both models, which are trivial to compute in GLM, but they aren't in GAM or SVR. The equivalent degrees of freedom for Support Vector Regression were introduced in the toolbox as defined in **??**.

- **PRSS[1], Variance-Normalized PRSS**

  Penalized Residuals Sum of Squares is introduced in the toolbox in order to provide a fit evaluation metric that penalizes the complexity of the predicted curve without requiring the degrees of freedom. However, PRSS is not suitable enough in the context of morphometric analysis, as it always provides better scores for target variables with low-variance and flat trends than for target variables with high-variance and nonflat trends, and that poses a problem as most of the voxels in the brain are 0-valued.

  For that reason a variance normalized version of the PRSS that takes into account this domain specific requirement and weights the score with the inverse of the variance of the predicted curve was introduced, the Variance Normalized Penalized Residuals Sum of Squares, which is formulated as $\frac{1}{\frac{1}{n-1}\sum_{i=1}^{n}(\hat{y}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i)^2}(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\int[f^{''}(x)]^2dx)$.

2.4 Interactive visualization tools

An interactive visualization tool is included in the toolbox to provide added insight on the results: it allows to load a 3D statistical map — generated with the fit evaluation method of choice — and several fitted models, and then plots the predicted curves of all the models for the voxel selected with the cursor, hence easing the task of inspecting the curves in the uncovered significant regions. An example of the aforementioned tool is found in **??**.

3 Implementation details

The whole toolbox has been implemented in Python. Numpy and scipy have been used for the numerical and scientific computing, NiBabel for handling the morphometric data in NIfTI format, scikit-learn for the machine learning algorithms and matplotlib and seaborn for plotting and for the visualization features.

4 Experiments

4.1 Dataset

5 Conclusions

Acknowledgments

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[1] Penalized Residual Sum of Squares