Feature selection and multivariate mapping in neuroimaging

CSML talk
September 2012

Jane Maryam Rondina
Features extraction in neuroimaging

• Individual voxels constrained to a specific brain area

• Summarization of anatomical regions of interest
Features extraction in neuroimaging

• Individual voxels in the whole brain

• Normalized images:
  – hundreds of thousands of voxels X few hundreds of examples;

• Challenges related to high dimensionality:
  – Model’s performance
  – Interpretation
Feature selection methods in neuroimaging

• Univariate selection
  – \textit{t}-test, \textit{F}-test, ANOVA

• Recursive Feature Elimination (RFE)\textsuperscript{1}
  – Recursive elimination of minimum weight voxel
  – Stepsize

• Searchlight\textsuperscript{2}

\textsuperscript{1} Guyon and Elisseefi, 2003
\textsuperscript{2} Kriegeskorte, Goebel and Bandettini, 2005
Stability selection

- It is a general theory to address problems related to variable selection or estimation of discrete structure (as graphs or clusters);
- It relies on data perturbation (e.g. sub-sampling) in combination with high-dimensional selection algorithms;
- Its properties are promising to applications involving high dimensional data (specially the $p>>n$ case)

Stability selection

\( \{ \hat{S}^\lambda; \; \lambda \in \Lambda \}, \) Variable selection in a traditional setting

Definition 1 (Selection probabilities) Let \( I \) be a random subsample of \( \{1, \ldots, n\} \) of size \( \lfloor n/2 \rfloor \), drawn without replacement. For every set \( K \subseteq \{1, \ldots, p\} \), the probability of being in the selected set \( \hat{S}^\lambda(I) \) is

\[
\hat{\Pi}_K^\lambda = P^*(K \subseteq \hat{S}^\lambda(I)).
\] (5)

Definition 2 (Stable variables) For a cutoff \( \pi_{thr} \) with \( 0 < \pi_{thr} < 1 \) and a set of regularisation parameters \( \Lambda \), the set of stable variables is defined as

\[
\hat{S}_{\text{stable}} = \{ k : \max_{\lambda \in \Lambda} \hat{\Pi}_k^\lambda \geq \pi_{thr} \}. \] (7)
Stability selection using LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) \(^1\)

\[
\hat{\beta}^\lambda = \arg\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_2^2 + \lambda \sum_{k=1}^{p} |\beta_k|,
\]

Randomised Lasso with weakness $\alpha \in (0, 1)$:

Let $W_k$ be i.i.d. random variables in $[\alpha, 1]$ for $k = 1, \ldots, p$. The randomised Lasso estimator $\hat{\beta}_{\lambda,W}$ for regularisation parameter $\lambda \in \mathbb{R}$ is then

$$\hat{\beta}_{\lambda,W} = \arg\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_2^2 + \lambda \sum_{k=1}^{p} \frac{|\beta_k|}{W_k}.$$ (13)
SCoRS (Survival Count on Random Subspace)

1. Input matrix (n examples, p voxels)
2. Split examples (training and test)
3. Select voxels randomly
4. Apply L1-norm regression in the voxels randomly selected
5. Increment counters for voxels with non-zero coefficient
6. Threshold
7. FS Binary mask
8. Compute Kernel Matrix
9. SVM (train and test)
10. Classification accuracy
11. Leave One (pair of subjects) Out
12. Iterations

Relevance map
Exploring parameters space in SCoRS

\[ S = \frac{p}{2^i \times n}, \text{ where } i = 4, 3, 2, 1, 0, -1, -2, -3, -4 \]  \hspace{1cm} (1)

\[ I = i \times r, \text{ where } i = 1:9 \text{ and } r \text{ was fixed as } 10^3 \]  \hspace{1cm} (2)

\[ T = i \times r, \text{ where } i = 1:9 \text{ and } r \text{ was fixed as } 10^{-1} \]  \hspace{1cm} (3)

S – subspace size
I – number of iterations
T - threshold
Exploring parameters space in SCoRS

Classification accuracy

Dataset 1

Dataset 2

Dataset 3

Exploring parameters space in SCoRS
(Number of features through threshold levels)

Dataset 1
B-SCoRS

Input matrix I
(n examples, p features)

Split examples
(training and test sets)

Select p features
randomly

Select a subset n1 from
training examples

Data matrix D
(n1 examples, p1 features)

Apply L1-norm regression
in D

Increment counter for
features with non-zero
coefficients

Threshold

Binary mask

SVM (train and test)

Classification results

Relevance map

LOO-CV
## Classification accuracy

<table>
<thead>
<tr>
<th></th>
<th>N features</th>
<th>TP</th>
<th>TN</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole brain</td>
<td>219727</td>
<td>0.63</td>
<td>0.70</td>
<td>0.67</td>
</tr>
<tr>
<td>No threshold</td>
<td>210922</td>
<td>0.63</td>
<td>0.70</td>
<td>0.67</td>
</tr>
<tr>
<td>T = 0.1</td>
<td>98738</td>
<td>0.67</td>
<td>0.73</td>
<td>0.70</td>
</tr>
<tr>
<td>T = 0.2</td>
<td>51094</td>
<td>0.63</td>
<td>0.73</td>
<td>0.68</td>
</tr>
<tr>
<td>T = 0.3</td>
<td>29958</td>
<td>0.63</td>
<td>0.73</td>
<td>0.68</td>
</tr>
<tr>
<td>T = 0.4</td>
<td>18046</td>
<td>0.63</td>
<td>0.80</td>
<td>0.72</td>
</tr>
<tr>
<td>T = 0.5</td>
<td>10704</td>
<td>0.67</td>
<td>0.80</td>
<td>0.74</td>
</tr>
<tr>
<td>T = 0.6</td>
<td>6170</td>
<td>0.67</td>
<td>0.77</td>
<td>0.72</td>
</tr>
<tr>
<td>T = 0.7</td>
<td>3265</td>
<td>0.67</td>
<td>0.77</td>
<td>0.72</td>
</tr>
<tr>
<td>T = 0.8</td>
<td>1473</td>
<td>0.67</td>
<td>0.73</td>
<td>0.70</td>
</tr>
<tr>
<td>T = 0.9</td>
<td>461</td>
<td>0.67</td>
<td>0.70</td>
<td>0.68</td>
</tr>
</tbody>
</table>

Data matrix: 219,727 voxels 240 examples
Nested CV for optimizing the number of features

Left 1 pair of subjects out of n pairs

Left 1 pair of subjects out of n-1 pairs

For each value of parameter in the range

**Run feature selection with n-2 pairs of subjects**

Train classifier with n-2 pairs of subjects

Test with the pair of subjects left out in the inner loop

Get the value of parameter which produced the best classification accuracy in the inner loop

Run feature selection with all pairs of subjects but the one left out in the outer loop

Train classifier

Test with the pair of subjects left out in the outer loop

<table>
<thead>
<tr>
<th>Method</th>
<th>N features</th>
<th>TP</th>
<th>TN</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCoRS</td>
<td>12006</td>
<td>0.67</td>
<td>0.77</td>
<td>0.72</td>
</tr>
<tr>
<td>RFE</td>
<td>32077</td>
<td>0.73</td>
<td>0.60</td>
<td>0.67</td>
</tr>
</tbody>
</table>
False positive selection
Spatial mapping
Thanks!