Tensor Ensemble Learning

Authors:
Ilia Kisil, Ahmad Moniri, Danilo P. Mandic

Electrical and Electronic Engineering Department
Imperial College London, SW7 2AZ, UK

E-mails: {i.kisil15, ahmad.moniri13, d.mandic}@imperial.ac.uk

November 28, 2018
Outline

- Wisdom of the crowd
- Ensemble learning and existing algorithms
- Multidimensional representation of data
- Basics of tensor decompositions
- Tensor Ensemble Learning (TEL)
- Simulations and results
Ensemble learning: The wisdom of the crowd
Ensemble learning: Motivation and limitations

- Every model has its own weaknesses $\Rightarrow$ Combining different models can find a better hypothesis
- Every model explores its own hypothesis space $\Rightarrow$ Robust to outliers
- Strong assumption that our individual errors are uncorrelated
Ensemble learning: Existing approaches

Bagging
Ensemble learning: Existing approaches

Bagging

Stacking
Ensemble learning: Existing approaches

**Bagging**

**Stacking**

**Boosting**
Tensors and basic sub-structures

$A(\cdot,\cdot,k)$

$A(:,j,:)$

$A(i,:,:)$

$I_1$

$I_2$

$I_3$

$I_1$

$I_2$

$I_3$

$I_1$

$I_2$

$I_3$

$I_1$

$I_2$

$I_3$

$I_1$

$I_2$

$I_3$

$I_1$

$I_2$

$I_3$

→ mode-1 unfolding

→ mode-2 unfolding

→ mode-3 unfolding
Multidimensional data and tensor construction

- Time intervals
- Subjects
- Frequency range

- 1,000 pixels
- 1,000 frames

- Coordinates
- Marker
- Time

- Raw signal
- Time domain
- Frequency domain

- Subject 1
- Subject 2
- Subject N

- One channel of interest, several subjects

Ilia Kisil, Imperial College London

Tensor Ensemble Learning
Tucker decomposition ⇔ HOSVD

Each vector of $A$ is associated with every vector of $B$ and $C$ through the core tensor $G$.

$$X \approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} G_{r_1r_2r_3} \cdot a_{r_1} \circ b_{r_2} \circ c_{r_3}$$

In general, the Tucker decomposition is not unique.

But the subspaces spanned by vectors of $A$, $B$, $C$ are unique.

By imposing orthogonality constraints on each factor matrix, we arrive at the natural generalisation of the matrix SVD, the higher-order SVD (HOSVD).
**Computation of the HOSVD**

1. Compute the factor matrices first:

\[
X^{(1)} = A \Sigma^{(1)} (V^{(1)})^T \\
X^{(2)} = B \Sigma^{(2)} (V^{(2)})^T \\
X^{(3)} = C \Sigma^{(3)} (V^{(3)})^T
\]  

(1)

2. Compute the core tensor

\[
G = X \times_1 A^T \times_2 B^T \times_3 C^T
\]

(2)

Where \(G = X \times_1 A^T \Leftrightarrow G^{(1)} = A^T X^{(1)}\)
Tensor ensemble learning (TEL): General concept

1. Apply tensor decomposition to each multidimensional sample to extract hidden information

2. Perform reorganisation of the obtained latent components

3. Use them to train an ensemble of base learners

4. For new sample, aggregate the knowledge about extracted latent components based on trained models in stage 3

Stage 1

\[ D : \left\{ (X^m, y^m) \right\}, \quad X^m \in \mathbb{R}^{I \times J \times K} \quad \text{for} \quad m = 1, \ldots, M \]

Tensor factorisation of each sample from \( D \)

\[ X^m = [G^m; A^m; B^m; C^m], \quad m = 1, \ldots, M \]

\[ A^m = [a^m_1 \ldots a^m_{R_a}]; B^m = [b^m_1 \ldots b^m_{R_b}]; C^m = [c^m_1 \ldots c^m_{R_c}] \]

Stage 2

Regrouping of the factor vectors into separate datasets

\[ D^A_1: \left\{ (a^m_1, y^m) \right\}; \ldots; D^A_{R_a}: \left\{ (a^m_{R_a}, y^m) \right\}, \quad m = 1, \ldots, M \]

\[ D^B_1: \left\{ (b^m_1, y^m) \right\}; \ldots; D^B_{R_b}: \left\{ (b^m_{R_b}, y^m) \right\}, \quad m = 1, \ldots, M \]

\[ D^C_1: \left\{ (c^m_1, y^m) \right\}; \ldots; D^C_{R_c}: \left\{ (c^m_{R_c}, y^m) \right\}, \quad m = 1, \ldots, M \]

Stage 3

Train classifier \( C_1(D^A_1) \)

Total number of classifiers \( N = R_a + R_b + R_c \)

Train classifier \( C_N(D^C_{R_c}) \)

Stage 4

For the \( X^{new} = [G^{new}; A^{new}; B^{new}; C^{new}] \), assign label \( y^{new} \) based on majority vote of \( \left\{ C_1(a^{new}_1), \ldots, C_N(c^{new}_{R_c}) \right\} \)
TEL: Formation of training set

\[ \mathcal{N} = (\mathbf{R}_a + \mathbf{R}_b + \mathbf{R}_c) \]
TEL: Formation of training set

\[ N = (R_a + R_b + R_c) \]
TEL: Formation of training set

\[ \text{Training set } 1 \quad \text{Training set } n \quad \text{Training set } N \quad \text{Label} \]

\[ a_1^{(1)} + b_{r_b}^{(1)} + c_{R_c}^{(1)} + \cdots + a_{R_a}^{(m)} + b_{R_b}^{(m)} + c_{R_c}^{(m)} = N \]

\[ N = (Ra + Rb + Rc) \]

Ilia Kisil, Imperial College London

Tensor Ensemble Learning

November 28, 2018 11 / 15
TEL: Formation of training set

\[ N = (R_a + R_b + R_c) \]
TEL: Formation of training set

\[ \text{Training set 1} \quad \text{Training set } n \quad \text{Training set } N \]
TEL: Formation of training set

\[ N = (R_a + R_b + R_c) \]
TEL: Training stage

ML algorithm: SVM, NN, KNN

Base learner 1
TEL: Training stage

ML algorithm: SVM, NN, KNN

Base learner 1

Base learner n
TEL: Training stage

ML algorithm: SVM, NN, KNN

Base learner 1

ML algorithm: SVM, NN, KNN

Base learner n

ML algorithm: SVM, NN, KNN

Base learner N
TEL: Testing stage

\[
\begin{bmatrix}
\hat{a}^{(\text{new})}_{R_a} \\
\hat{b}^{(\text{new})}_{R_b} \\
\hat{c}^{(\text{new})}_{R_c}
\end{bmatrix} =
\begin{bmatrix}
\hat{a}^{(\text{new})}_1 \\
\hat{b}^{(\text{new})}_1 \\
\hat{c}^{(\text{new})}_1
\end{bmatrix} + \cdots +
\begin{bmatrix}
\hat{a}^{(\text{new})}_n \\
\hat{b}^{(\text{new})}_n \\
\hat{c}^{(\text{new})}_n
\end{bmatrix}
\]

Ilia Kisil, Imperial College London
TEL: Testing stage

\[
\begin{align*}
\mathbf{b}_1^{(\text{new})} + \cdots + \mathbf{b}_{R_b}^{(\text{new})} &= \mathbf{c}_1^{(\text{new})} \\
\mathbf{a}_1^{(\text{new})} + \cdots + \mathbf{a}_{R_a}^{(\text{new})} &= \mathbf{c}_c^{(\text{new})}
\end{align*}
\]

Predicted label

Total vote (probability based)

<table>
<thead>
<tr>
<th></th>
<th>Score</th>
<th>Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple (1)</td>
<td>0.70</td>
<td></td>
</tr>
<tr>
<td>Car (2)</td>
<td>0.20</td>
<td></td>
</tr>
<tr>
<td>Tomato (3)</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>

Ilia Kisil, Imperial College London
Tensor Ensemble Learning
November 28, 2018 13 / 15
TEL: Testing stage

\[ a_1^{(\text{new})} + b_1^{(\text{new})} + \cdots + c_{R_c}^{(\text{new})} = c_1 \]

Predicted label

Base Learner 1

Base Learner n

Base Learner N

<table>
<thead>
<tr>
<th>Score</th>
<th>Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple (1)</td>
<td>4.90</td>
</tr>
<tr>
<td>Car (2)</td>
<td>4.90</td>
</tr>
<tr>
<td>Tomato (3)</td>
<td>2.20</td>
</tr>
</tbody>
</table>

Ilia Kisil, Imperial College London
Tensor Ensemble Learning
November 28, 2018 13 / 15
TEL: Testing stage

\[ a_1^{(\text{new})} + b_1^{(\text{new})} + \cdots + c_{R_c}^{(\text{new})} = e_{R_c}^{(\text{new})} \]

Predicted label

<table>
<thead>
<tr>
<th>Score</th>
<th>Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple (1)</td>
<td>5.60</td>
</tr>
<tr>
<td>Car (2)</td>
<td>9.80</td>
</tr>
<tr>
<td>Tomato (3)</td>
<td>3.60</td>
</tr>
</tbody>
</table>

Total vote (probability based):

\[
\begin{bmatrix}
0.7 \\
0.2 \\
0.1 \\
0.1 \\
0.4 \\
0.5 \\
0.1 \\
\end{bmatrix}
\]
ETH-80 dataset

- 8 different categories
- 10 objects per category
- Each object is captured under 41 different viewpoints
- Total: 3280 samples
Performance of base estimators

- We employed HOSVD with multilinear rank \((5, 5, 2)\)
- Utilised 12 base estimators (train/test split 50%)
- None of the base classifiers exhibited strong performance on the training set
- Combinational behaviour is similar to classic ensemble learning
Comparison of the overall test performance

- Random split into training and test data varied in range from 10% to 70%
- Hyperparameter tuning: grid search with the 5-fold CV of the training data
- For fair comparison, the Bagging classifier also utilised 12 base learners
Conclusions: Key points to take home

1. TEL is a novel framework for generating ensembles of base estimators for multidimensional data

2. TEL highly parallelisable and suitable for large-scale problems

3. Enhanced performance is due to ability to obtain uncorrelated surrogate datasets that are generated by HOSVD
Conclusions: Key points to take home

1. TEL is a novel framework for generating ensembles of base estimators for multidimensional data

2. TEL highly parallelisable and suitable for large-scale problems

3. Enhanced performance is due to ability to obtain uncorrelated surrogate datasets that are generated by HOSVD

New Software: Higher Order Tensors ToolBOX (HOTTBOX)

- Our python package for multilinear algebra: github.com/hottbox/hottbox
- Documentation: hottbox.github.io
- Tutorials: github.com/hottbox/hottbox-tutorials
Thank you for your attention 😊

Questions?