PROGRESS IN EMULATION FOR RADIATIVE TRANSFER MODELING AND MAPPING

Jochem Verrelst, Juan Pablo Rivera, Pablo Morcillo, Maria Lumbieres, Jorge Vicent & Jose Moreno

Image Processing Laboratory, University of Valencia (Spain)
Any difference? Which model would you choose?

SCOPE (RTM)

Emulator (emulated SCOPE)
BACKGROUND

Advanced RTMs: generation of a large LUT (>1000#)

SCOPE

DART

MODTRAN

Advanced RTMs: more realistic but slow
Emulators are statistical models that approximate the processing (input-output) of a physical model (e.g. RTM) - at a fraction of the computational cost:

making a statistical model from a physical model
Regression vs. Emulation:

Common use of machine learning in optical RS:

Statistical regression method:
- Variable/data-driven, black box, 1 output, portability is questionable

Emulation in optical RS:

Replace RTM:
- Multiple applications, e.g. inversion
  - Radiometric method: Spectral fitting
  - Portable: generally applicable

“spectral redundancy” is a blessing
Input (variables + spectra) → Splitting into training/validation → PCA on spectra → MLRA training looping over components → Prediction of components → Reconstruction of spectra → Validation → Emulator

**Processing steps emulation**

- **PCA on spectra**
  - \[ S_c = U \cdot X \]
  - [Original spectra graph]
  - [PCA components graph]

- **MLRA training looping over components**
  - \[ W = (Y + \lambda I)^{-1} \cdot S_c \]

- **Prediction of components**
  - \[ S_p = S_c \cdot W \]
  - [Predicted spectra graph]

- **Reconstruction of spectra**
  - \[ X_r = U^T \cdot S_p \]
With ARTMO’s emulation processing chain any RTM can be converted into an emulator.

\[ X_r = U^T \cdot S_p \]

\[ \text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_{i,\text{pred}} - y_{i,\text{exp}})^2} \]

\[ \text{NRMSE} = \frac{1}{Y_{\text{max}} - Y_{\text{min}}} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2} \]


http://ipl.uv.es/artmo/
Emulators great idea... what about accuracy?

1) Role of machine learning regression algorithm?

2) Role of dimensionality reduction (DR) method?

3) Role of LUT size training?

4) Role of data type?

If OK with losing some accuracy, various applications are opened:

Fast RTM output generation
1. Fast scene generation
2. Fast global sensitivity analysis
3. Fast inversion

Latest developments in Emulator toolbox:

• New ML algorithms implemented:
  1. Multi-output support vector regression
  2. Multi-output Gaussian process regression

  In total 9 ML algorithms implemented in toolbox.

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>Algorithm Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forests (TreeBagger) (RF TB)</td>
<td>Kernel Ridge Regression (KRR)</td>
</tr>
<tr>
<td>Random Forests (Fitensemble) (RF FE)</td>
<td>Gaussian Process Regression (GPR)</td>
</tr>
<tr>
<td>Neural Network (NN)</td>
<td>Multi-output GPR (MGPR)</td>
</tr>
<tr>
<td>Support Vector Regression (SVR)</td>
<td>Varational Heteroscedastic GPR (VH GPR)</td>
</tr>
<tr>
<td>Multi-output SVR (MSVR)</td>
<td></td>
</tr>
</tbody>
</table>

Experimental setup:

• PROSAIL 1500 LHS simulations, all variables ranged
• 20 PCA and 70/30 training-testing data splitting
Performances emulators: accuracy & CPU

GPR very accurate (<2%)

![Graph showing NRMSE % vs Wavelength (nm) for different emulators: GPR, VH.GPR, MSVR, RF (TB), KRR, SVR, NN, MGPR, RF (FS). The graph includes a table showing training time in seconds for each emulator: GPR 179, VH.GPR 724, MSVR 52, RF (TB) 32, KRR 4, SVR 2106, NN 42, MGPR 12, RF (FS) 21. Testing time is <1 s.]
Validation GPR emulator

Overview statistics emulator vs validation

- Mean and SD perfectly matching.
- Some small mismatch in boundaries.

Running GPR emulator:

1.0 s
Applications (1/3)

Scene generation
Emulation for Scene generation

- GPR emulator applied for scene generation
- Compared against RTM scenes: PROSAIL & SCOPE

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Units</th>
<th>Class 1</th>
<th>Class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaf area index</td>
<td>LAI</td>
<td>m²/m²</td>
<td>Uniform: 3 - 6</td>
<td>Uniform: 0 - 2</td>
</tr>
<tr>
<td>Chlorophyll a+b content</td>
<td>$C_{ab}$</td>
<td>μg/cm²</td>
<td>Uniform: 20 - 60</td>
<td>Uniform: 10 - 35</td>
</tr>
</tbody>
</table>

See Talk TU4.R9.5 for emulation of real scenes
Applications (2/3)

Global Sensitivity Analysis (GSA)

GSA techniques quantify the relative importance of each input parameter to model outputs.
Emulators applied into GSA: PROSPECT-4

ARTMO’s GSA toolbox

PROSPECT-4 (122 s)

KRR (55 s)  NN (55 s)  GPR (58 s)

1000#/variable
PROSAIL

4SAIL (103 s)

Emulator GPR (32s)

Emulator KRR (28s)

Emulator VHGPR (37s)

X 3

1000#/variable
Applying emulation to $L_{TOA}$ (1/3)

50 PROSAIL spectra

50 6S spectra

\[ L_{TOA} = L_0 + \frac{T_{gas} \cdot E_{tot} \cdot T_{tot} \cdot \rho}{\pi (1 - S \rho)} \]

2500 TOA radiance spectra

Emulator

- 30 PCA
- 70/30% train/test
$L_{TOA}$ Emulator results (2/3)

**GPR**

**KRR**

**NN**

Validation

NRMSE %

Wavelength [nm]

GPR  
KRR  
NN
• Strong influence of AOT in visible.
• Strong influence of water absorption regions
• SWIR seems attractive for TOA retrieval of vegetation properties (hardly atmospheric effects).
Applications (3/3)

Inversion
Emulators into numerical inversion

Spectral fitting:
Minimization algorithm: lsqnonlin

Output maps of RTM variables

RTM

Per-pixel RTM iterations: very slow method, inapplicable to advanced RTMs.

Emulation of an RTM
DART KRR emulator applied to HyPlant DUAL (450-2500 nm) < 1 h

SCOPE KRR emulator applied HyPlant DUAL (bare soil spectra added)

CC  LCC  APAR  LAI  CCC (LCC x LAI)

LAI  RMSE  fAPAR  CWC (Cw x LAI)  RMSE

Forest

Agriculture

PROSAIL: 66 min

NN emulator: 21 min

X 3.1

23/24
Take home messages

Emulation approximates physical models with sufficient accuracy and tremendous gain in speed.

- Emulation permits fast rendering of optical images
- Emulation permits fast calculation of global sensitivity analysis
- Emulation permits fast numerical inversion of RTM against an image for biophysical variables retrieval
Thanks!

More talks about emulation:

1) Jorge Vicent: TU1.R7.4: STATISTICAL LEARNING FOR END-TO-END SIMULATIONS (09:30 - 09:50)
3) Daniel Heestermans: WE2.R7.5 MULTIOUTPUT AUTOMATIC EMULATOR FOR RADIATIVE TRANSFER MODELS (12:30-12:50)