STIC-AmSud 2\textsuperscript{nd} Meeting

Curitiba, Oct. 23, 2014

Prof. Laurent Heutte

Laurent.Heutte@univ-rouen.fr
http://www.litislab.eu/Members/lheutte
University of Rouen, France

- Located in the north of Paris (100 km)
- 33000 students
- 7 Faculties (research/teaching)
  - Medicine
  - Sciences
  - Literacy
  - Law
  - Technology
  - Economic sciences
  - Psychology
Faculty of Sciences and Techniques

- 3300 students
- 66 diplomas
- 400 professors and researchers
- 200 administrative staff
- Faculty divided into:
  - 7 departments (teaching): Computer Science, Computer Engineering, Physics, Biology, Mathematics,…
  - 14 laboratories (research): LITIS, CORIA, IRCOF, … some may associated with CNRS, INRIA, INSERM,…
LITIS Lab. (http://www.litislab.eu)

- Laboratory of Computer Science, Information Processing and Systems
- Depending on 3 organizations located in Upper Normandy: University of Rouen, University of Le Havre, INSA Rouen
- Scope: Sciences and Technology of Information and Communications
  - All formal and practical aspects of information processing
- 90 faculty members (whose 31 Prof, 5 Assoc. Prof., 54 Ass. Prof.)
- 7 research teams
- 80 PhD students and post-doc
- Head of the Lab: Prof. T. Paquet
Document and Learning Team

- Head: Prof. L Heutte
- 16.5 staff members – 7 PR, 9.5 Ass. Prof.
- 4 post-doc and engineers
- 16 PhD students
Scientific Issues

- Machine Learning and Pattern Recognition

- Joint learning of representations and decisions
  - Dictionary learning and variable selection, deep learning
  - Kernel learning (SVM, Kernel PCA, SimpleMKL, regularization path)
  - Graphs and learning (isomorphism, classification, …)
  - Model selection, bayes estimators and risks

- Model adaptability
  - Markovian models, multi-streams HMM, structure adaptation, Markov random fields and CRF
  - Learning with unknown or evolutive costs, multi-objective learning, hyper-parameters in classifier ensembles (random forests, DRF, one-class)
  - Multi-task learning
Application domains

- Access to information
  - Handwriting recognition
  - Spotting
  - Information extraction
  - Complex manuscripts
  - Digital libraries

- Recommendation systems

- Biomedical information processing
  - Brain Computer interface
  - Analysis of motor control data

- Medical image classification
- Medical image segmentation
STIC-AmSud French Team

- Prof. L. Heutte, PhD, PhD supervisor
  - Off-line and on-line handwriting analysis and recognition
  - Handwritten document analysis (bank checks, postal addresses, incoming mails, old manuscripts)
  - Information extraction and retrieval in handwritten documents
  - Classifier ensemble learning, classifier selection in ensembles
  - Pattern spotting, document image retrieval

- Ass. Prof. Caroline Petitjean, PhD
  - Medical image analysis, segmentation and classification
  - Cardiac MRI image segmentation with shape prior (graph-cut)
  - Medical image modelling
  - Image retrieval

- Ass. Prof. Simon Bernard, PhD
  - Classifier ensemble learning
  - Random forests
Random Forests

- A joint work with Simon BERNARD (PhD)
  - Understand why and how performance is affected by hyper-parameters
  - Improve Random Forest induction procedure
  - Forest-RK, Dynamic Random Forests

- A joint work with Chesner DESIR (PhD), Caroline PETITJEAN and Simon BERNARD
  - A new challenging application: endomicroscopic image classification
  - Optimizing Extra-Trees and random subwindows for medical image classification
  - One-Class Random Forest
Ensemble of Classifiers

IDEA:
- Do not learn a *single classifier* but learn a *set of classifiers*
- *Combine the predictions* of multiple classifiers

MOTIVATION:
- *Reduce variance*: results are less dependent on peculiarities of a single training set
- *Reduce bias*: a combination of multiple classifiers may learn a more expressive concept class than a single classifier

KEY STEP:
- Building an ensemble of *diverse* classifiers from a single training set
Building an Ensemble of Classifiers

- By manipulating training data
  - cross-validated committees
  - bagging
  - Boosting

- By manipulating input features
  - adding noise
  - random subspace method

- By manipulating output features
  - error-correcting output codes
Random Forests [Breiman, 2001]

**Definition**

A random forest is a classifier consisting of a collection of tree-structured classifiers

\[
\{ h(x, \Theta_k), \ k = 1, ..., L \}
\]

where the \( \{\Theta_k\} \) are independent identically distributed random vectors.
Forest-RI [Breiman, 2001]

Two randomization principles:

- Bagging
  - Random sampling with replacement of $N$ training instances to form the boostra samples $T_k$, for $k = 1..L$
  - the $h_k$ classifier is trained on $T_k$
  - aggregation of the $L$ resulting classifiers for the prediction

$T\{x_1, x_2, \ldots, x_N\}$

$h(x) = y_i$
Two randomization principles:

\[ \{A_1, A_2, \ldots, A_M\} \]

\[
\frac{T}{A_i} \leq \alpha_i^{opt}
\]

Random Feature Selection (RFS):
- randomly choose \( K \) features among \( M \)
- built the best splitting rule from these features
Random Forests

- Random Forests are efficient...
  - Variable selection/variable importance (efficient, even in high dimensional feature space)
    - [Breiman, 2001], NIPS 2003 Workshop on Feature Extraction
  - "outlier" detection, proximity measures
  - Forest-RI comparable to AdaBoost
    - [Breiman, 2001, Baneld et al., 2004, Boinee et al., 2005, Geurts et al., 2006, Baneld et al., 2006]

- But this efficiency is not always easy to reach in practice...
An example...

- Madelon dataset (UCI repository, 2600 samples, 500 features, 2 classes)
  - Forest-RI : $K = 22 \ (\sqrt{M})$, $L = 300$ => test error rate = 30.50%
  - Forest-RI : $K = 260$, $L = 300$ => test error rate = 17.73%
  - Forest-RI : $K = 260$, $L = 100 \ ($selected trees$)$ => test error rate = 15.96%

- Understand why and how performance is affected by the hyper-parameters $K$ and $L$
- Improve Random Forest induction procedure
- => PhD work of Simon Bernard
Influence of K: strength of randomization

- K controls the strength of randomization in the tree induction
  - Integer between 1 and M
  - K=1 => maximum randomization
  - K=M => no randomization (like CART, C4.5…)

- How to choose K?
  - Arbitrary values of K
  - K = 1, [Beiman 2001]
  - K = log₂(M)+1, [Breiman 2001]
  - K = sqrt(M), [Geurts et al, 2006]

- Experiments on 14 UCI datasets with exhaustive search of K*
  - K* vs K₁, K_sqrt, K_log => K_sqrt is the closest to K* on average…
  - … but sometimes K_sqrt is a really bad choice!
Forest-RK: a new push-button algorithm

- Same as Forest-RI, i.e. with bagging and RFS…
- … but K is chosen randomly at each node
- => K is not an hyper-parameter anymore

- Experiments conducted on 14 UCI datasets [MCS2009]
  ✓ Forest-RK close to K* and at least as good as Ksqrt in majority

- Why does it work?
  ✓ K acts as a trade-off for balancing performance and diversity
  ✓ few relevant features => RFS deteriorates the "performance/diversity" trade-off
  ✓ lot of relevant features => weakens the RFS effects in split selection

Bernard et al. Influence of hyperparameters on random forest accuracy. MCS 2009
Influence of L: tree selection in RF

Idea: Generate sub-forests and study their performance

\[ \text{error rates with respect to the number of trees in sub-forests} \]

→ For 18 of the 20 datasets: at least one sub-forest significantly better
→ Sub-forests at least comparable to the 300-trees RF, with sometimes less than 10% of trees

Dynamic Random Forests

- Main idea: guide the tree induction
  - make the new tree dependent on the subforest already built
  - Boosting-like procedure

Training instances weighting:
→ increase weights for wrongly classified instances
Dynamic Random Forests

Test error rates with respect to the number of trees

- 500-trees forests
- DRF significantly outperforms Forest-RI/\sqrt{K} for 14 datasets
- DRF > Forest-RI/\sqrt{K^*} > Forest-RI/\sqrt{K}

One-Class Classification

One-class classification (OCC) consists in:

1. learning from only one class of objects (the target class, e.g. medical abundant healthy cases) ...

2. ... and discriminating this class of interest from one or several other classes of objects (the outlier class), with no prior knowledge about the outlier class (e.g. rare pathological cases)

OCC is also called “novelty detection”, “anomaly detection”, “outlier detection” or “data description”, often depending on applications.
**Numerous applications:**

- typist recognition, authorship verification, intrusion detection, mobile-masquerader detection, machine or structure health monitoring, fault detection...
- in the medical field: pathological samples are rare and therefore not reliable enough to train a binary classifier
- one example is the diagnosis of lung diseases via alveoscopy:

![Healthy and Pathological Images](image-url)

- Healthy
- Pathological

(a) Non-smoking subjects  
(b) Smoking subjects
Application in lung disease diagnosis

Classification of endomicroscopic images of the lung [MLMI 2012]:

- A new technique → new images → uncertain oracle on pathological images

Learning from healthy images only → One-Class Paradigm
OCC: competitive algorithms (1)

- Experiments and results (cont'd)

Natural choice for OCC [KM10]: Density estimators, based on estimating density function of target data: Mixture of gaussians, Parzen windows

Main issues with density approaches:

- Rarely effective for high dimensional problems, requiring large sample sizes for reliable estimation ("curse of dimensionality")
- Threshold on probability outputs chosen beforehand
OCC: competitive algorithms (2)

Discriminative approaches, based on implicit decision frontier between classes to discriminate with

- modification of inner workings of existing algorithms (SVDD, OCC-SVM)
  - SVDD [TD04] minimizes the volume of an hypersphere around the target class.
  - OCC-SVM [SPST+01] separates the target from the origin with maximal margin with an hyperplane.

- generation of artificial, uniformly distributed, outliers
  - hypercube of side $c$ in $M$-sized feature space
  - requires an exponential amount of data to cover the whole feature space: unusable;

\[
V_{\text{hypercube}}(c) = c^M
\]

\[
N_{\text{outliers}}(c) = \frac{V_{\text{hypercube}}}{V_{\text{outlier}}} = \frac{c^M}{(10^{-p})^M}
\]
One-Class Random Forest

Our solution:
- Using classifier ensemble paradigm to break the curse of dimensionality, by subsampling the feature space and the training set.
- Generating more outlier data in sparse target regions, and less in densely populated target regions.

Our approach: One-Class Random Forest (OCRF), combining ensemble learning principles from traditional Random Forest algorithm with an original outlier generation method.

OCRF is composed of three main steps:
(i) extraction of **density information** from whole target data
(ii) generation of outlier data in **bootstrap** samples projected into **RSM** subspaces
(iii) induction of a **Forest-RI** on augmented dataset: bagging + RFS
OCRF: framework illustration
**OCR-F algorithm**

**Require:** Training set $T$, $N_{outlier}$, $\Omega_{outlier}$, $L$ individual trees, $K_{RSM}$

**Ensure:** A random forest classifier

1. **(A) Density information extraction**
   2. Compute $H_{\text{target}}$, normalized histogram of target data in $T$
   3. Compute $H_{\text{outlier}}$, normalized histogram of generated outlier data, complementary of $H_{\text{target}}$, i.e. $H_{\text{outlier}} = 1 - H_{\text{target}}$

4. **(B) Outlier generation and forest induction**
   5. **for** $l = 1$ **to** $L$ **do**
   6. (i) Draw a bootstrap sample $T_l$ from training set $T$
   7. (ii) Project $T_l$ onto a random subspace of dimension $K_{RSM}$
   8. (iii) Generate $N_{outlier}$ outlier data according to $H_{\text{outlier}}$ in the domain $\Omega_{outlier}$
   9. (iv) Train a standard decision tree on the augmented dataset
10. **end for**
11. **return** random forest model
OCRF algorithm

Advantages:

- full description obtained from whole dataset, instead of subsamples (i)
- far less artificial outlier data generated with RSM feature space sampling (ii) with a generation process adapted to the dataset (i)
- handling of high dimensional problems (ii & iii)
- general purpose framework, best or competitive results obtained with random-forest principles (iii): bagging + RSM + RFS

Main parameters:

- Number of outliers to generate in (ii)
- Domain of outlier generation, e.g. surrounding hyperbox
- Forest parameters: L, $K_{RSM}$, $K_{RFS}$
OCR experiments

- Results on 78 UCI datasets

<table>
<thead>
<tr>
<th></th>
<th>OCRF</th>
<th>OCSVM</th>
<th>Gauss</th>
<th>Parzen</th>
<th>Mog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. rank</td>
<td>2.4 ± 1.1</td>
<td>4.0 ± 1.3</td>
<td>1.9 ± 1.15</td>
<td>3.8 ± 1.1</td>
<td>2.8 ± 1.0</td>
</tr>
</tbody>
</table>

(a) Mean rank values

<table>
<thead>
<tr>
<th>row &gt; col</th>
<th>OCRF</th>
<th>OCSVM</th>
<th>Gauss</th>
<th>Parzen</th>
<th>Mog</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCRF</td>
<td>-</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>OCSVM</td>
<td>-</td>
<td></td>
<td>-1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>Gauss</td>
<td>-</td>
<td></td>
<td></td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>Parzen</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td>-1</td>
</tr>
</tbody>
</table>

(b): OCRF > {OCSVM, Parzen}, nothing can be said when compared to {Gauss} or {MoG}

Désir et al. One-Class Random Forests. PR, 46(12), 2013
Publications

- **On Random Forest**
Publications

- On One-Class Random Forests

- On application of Random Forests to Medical Image Classification
Discussion...