Some Hybrid Predictive Models in the interface of Statistics & Machine Learning

by

Dr. Tanujit Chakraborty

Center for Data Sciences

International Institute of Information Technology Bangalore, India.

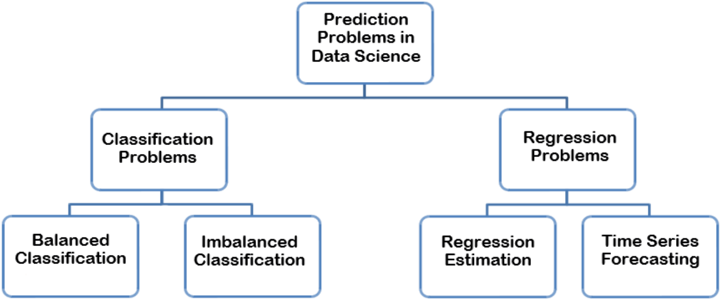
Outline of the talk

- Motivation
- Preliminaries
- Imbalanced Classification Problem in Software Defect Prediction
- Regression Estimation Problem in Process Efficiency Improvement
- Conclusions and Future Challenges

Part I: Motivation

Motivation

- "**Prediction** is very difficult, especially if it's about the future" - Niels Bohr, Father of Quantum Mechanics.
- Predictive modelling approaches are used in the fields of statistics and machine learning, mainly for their accuracy and ability to deal with complex data structures.
- In this talk, I will discuss about some novel Hybrid Predictive models motivated by the applied problems from the domain of Quality Control (regression estimation) and Software Defect Prediction (imbalanced classification).



Motivation

- Primary motivation comes from the real-world data sets, with a variety of data types, such as business, process efficiency improvement, water quality control, and software defect prediction, among many others.

- As a secondary motivation, we emphasis on the development of hybrid models that are scalable (the size of the data does not pose a problem), robust (work well in a wide variety of problems), accurate (achieve higher predictive accuracy), statistically sound (have desired asymptotic properties), and easily interpretable.

- The newly developed hybrid methods are shown to outperform the current state-of-the-arts and overcome the deficiencies of the hybrid models available in the literature.

- Both theoretical (asymptotic results) and computational aspects of the proposed hybrid frameworks are studied.

Brief History of Prediction Models

- Linear Regression (Galton, 1875).
- Linear Discriminant Analysis (R.A. Fisher, 1936).
- Logistic Regression (Berkson, JASA, 1944).
- k-Nearest Neighbor (Fix & Hodges, 1951).
- Parzen's Density Estimation (E Parzen, AMS, 1962)
- ARIMA Model (Box and Jenkins, 1970).
- Classification and Regression Tree (Breiman et al., 1984).
- Artificial Neural Network (Rumelhart et al., 1985).
- MARS (Friedman, 1991, Annals of Statistics).
- SVM (Cortes & Vapnik, Machine learning, 1995)
- Random forest (Breiman, 2001).
- Deep Convolutional Neural Nets (Krizhevsky, Sutskever, Hinton, NIPS 2012).
- GAN (Goodfellow et al., NIPS 2014).
- Deep Learning (LeCun, Bengio, Hinton, Nature 2015).
- Bayesian Deep Neural Network (Y. Gal, Islam, Zoubin, ICML 2017).

Need for Hybridization

- Statistical issue: It is often the case that the model space is too large to explore for limited training data, and that there may be several different models giving the same accuracy on the training data. The risk of choosing the wrong model can be reduced by combining two models, like CART and ANN.
- Representation issue: In many learning tasks, the true unknown hypothesis could not be represented by any hypothesis in the hypothesis space. By hybridization, it may be possible to expand the space of representable functions. Thus the learning algorithm may be able to form a more accurate approximation to the true unknown hypothesis.
- Computational issue: Many learning algorithms perform some kind of local search that may get stuck in local optima. Even if there are enough training data, it may still be challenging to find the best hypothesis. By combining two or more models, the risk of choosing a wrong local minimum can be reduced.

Ensemble & Hybrid Models

- Problem: Single models have the drawbacks of sticking to local minimum or over-fitting the data set, etc.
- Ensemble models are such where predictions of multiple models are combined together to build the final model.
- Examples: Bagging, Boosting, Stacking and Voting Method.
- Caution: But ensembles don't always improve accuracy of the model but tends to increase the error of each individual base classifier.
- Hybrid models are such where more than one models are combined together.
- It overcomes the limitations of single models and reduce individual variance & bias, thus improve the performance of the model.
- Caution: To build a good ensemble classifier the base classifier needs to be simple, as accurate as possible, and distinct from the other classifier used.
- Desired: Interpretability, Less Complexity, Less Tuning Parameters, high accuracy.

Popular Hybrid Prediction Models

- Perceptron Trees (Utgoff, AAAI, 1988).
- Entropy Nets (Sethi, Proceeding of IEEE,1990).
- Neural trees (Sirat & Nadal, Network, 1990).
- Sparse Perceptron Trees (Jackson, Craven, NIPS, 1996).
- SVM Tree Model (Bennett et al., NIPS, 1998)
- Hybrid DT-ANN Model (Jerez-Aragones et al., 2003, AI in Medicine)
- Flexible Neural Tree (Chen et al., Neurocomputing, 2006)
- Hybrid DT-SVM Model (Sugumaran et al,, Mechanical Systems and Signal Processing, 2007).
- Hybrid CNN–SVM Classifier (Niu et al., PR, 2012).
- Hybrid DT model utilizing local SVM (Dejan et al., IJPR, 2013).
- Neural Decision Forests (Bulo, Kontschieder, CVPR, 2014).
- Deep Neural Decision Forests (Kontschieder, ICCV, 2015).
- Soft Decision Tree (Frosst, Hinton, Google AI, 2017).
- Deep Neural Decision Trees (Humbird et al., IEEE TNNLS, 2018).
- Adaptive Neural Trees (Tanno et al. ICML, 2019).

Some Drawbacks of the Existing Hybrid Models

- Theoretical Robustness: Regardless of the practical use of SDT and neural trees, theoretical properties like universal consistencies of these hybrid methods are unknown. Thus, one needs to analyze the data complexity for splitting, which leads to more accurate classification in the neural trees node.

- High-dimensional set-up: Accurate classification of high dimensional feature space leads to more depth trees, thus achieving less depth neural trees require more complex computations at each node.

- Small Sample Size and Interpretability: The previously used hybrid models sometimes over-fit for small or moderate sample-sized data sets. In DNDT, each node in their oblique decision tree involves all features rather than a single feature, which renders the model uninterpretable.

Part II: Preliminaries

Decision Trees

- Decision tree is defined by a hierarchy of rules (in form of a tree).
- Rules from the internal nodes of the tree are called root nodes.
- Each rule (internal node) tests the value of some feature.
- Labeled training data is used to construct the Decision tree. The tree need not to be always a binary tree.
- CART is a greedy divide-and-conquer algorithm.
- Attributes are selected on the basis of an impurity function (e.g., IG for Classification & MSE for Regression).
- CART (Breiman et al., 1984), RF (Breiman, 2001 at ML), Random Survival Forest (Ishwaran et al., 2008 at Ann. App. Stat.) and BART (Chipman et al., 2010 at Ann. App. Stat).
- **Pros:** Built-in feature selection mechanism, Comprehensible, easy to design, easy to implement, good for structural learning.
- **Cons:** Too many rules loose interpretability, risk of over-fitting, sticking to local minima.

Artificial Neural Networks

- ANN is composed of several perceptron-like units arranged in multiple layers.
- Consists of an input layer, one or more hidden layer, and an output layer.
- Nodes in the hidden layers compute a nonlinear transform of the inputs.
- **Universal Approximation Theorem (Hornik, 1989)**: A one hidden layer FFNN with sufficiently large number of hidden nodes can approximate any function.
- **Pros:** Able to learn any complex nonlinear mapping or approximate any continuous function.
- **Pros:** No prior assumption about the data distribution or input-output mapping function.
- **Cons:** When applied to limited data can overfit the training data and lose generalization capability
- **Cons:** Training ANN is time-consuming and selection of the network topology lack theoretical background, often "trial and error" matter.

Statistical Learning Theory: Consistency

- Statistical learning theory (SLT) studies mathematical foundations for machine learning models, originated in late 1960s.
- Basic concept of Consistency: A learning rule, when presented more and more training examples → the optimal solution.

Definition (Consistency)

Given an infinite sequence of training points $(X_i, Y_i)_{i \in N}$ *with* μ*. For each* $n \in N$*, let* f_n *be a classifier for the first* n *training points. The learning algorithm is called consistent with respect to* μ *if the risk* $R(f_n)$ *converges to the risk* $R(f_{Bayes})$*, that is for all* $\epsilon > 0$,

$$\mu(R(f_n) - R(f_{Bayes}) > \epsilon) \to 0 \text{ as } n \to \infty.$$

Definition (Universally Consistency)

The learning algorithm is called universally consistent if it is consistent for all probability distributions μ*.*

Statistical Learning Theory in Decision Trees & Neural Networks

- Consistency of data driven histogram methods (Lugosi & Nobel, 1996, Annals of Statistics).
- Generalization Bounds for Decision Trees (Mansour et al., 2000, COLT).
- Consistency of random survival forests (Ishwaran et al., 2010, SPL).
- Consistency of Online Random Forest (Denil et al., 2013, ICML).
- Consistency of Random Forest (Scornet et al., 2015, Ann. Stat.).
- Posterior concentration for Bayesian regression trees and forests (Rockova et al., 2020, Ann. Stat.).

- Strong Universal Consistency of FFNN Classifier (Lugosi & Zeger 1995, IEEE Information Theory).
- Approximation properties of ANN (Mhaskar, 1993, Advances in Computational Mathematics).
- Prediction Intervals for Artificial Neural Networks (Hwang, Ding, 1997, JASA)
- On Deep Learning as a remedy for the curse of dimensionality (Bauer & Kohler, 2019, Ann. Stat.).
- Consistent Sparse Deep Learning (Sun et al., 2021, JASA).

Consistency of data-driven histogram methods for density estimation and classification

Theorem (Lugosi and Nobel, 1996, Annals of Statistics)

Let $(\underline{X}, \underline{Y})$ be a random vector taking values in $\mathbb{R}^p \times C$ and L be the set of first n outcomes of $(\underline{X}, \underline{Y})$. Suppose that Φ is a partition and classification scheme such that $\Phi(L) = (\psi_{pl} \circ \phi)(L)$, where ψ_{pl} is the plurality rule and $\phi(L) = (L)_{\tilde{\Omega}_n}$ for some $\tilde{\Omega}_n \in \mathcal{T}_n$, where $\mathcal{T}_n = \{\phi(\ell_n) : P(L = \ell_n) > 0\}$. Also suppose that all the binary split functions in the question set associated with Φ are hyperplane splits. As $n \to \infty$, if the following regularity conditions hold:

$$\frac{\lambda(\mathcal{T}_n)}{n} \to 0 \tag{0.1}$$

$$\frac{log(\triangle_n(\mathcal{T}_n))}{n} \to 0 \tag{0.2}$$

and for every $\gamma > 0$ and $\delta \in (0, 1)$,

$$\inf_{S \subseteq \mathbb{R}^p : \eta_X(S) \geq 1-\delta} \eta_x(x : diam(\tilde{\Omega}_n[x] \cap S) > \gamma) \to 0 \tag{0.3}$$

with probability 1. then Φ is risk consistent.

Eqn. (0.1) is the sub-linear growth of the number of cells, Eqn. (0.2) is the sub-exponential growth of a combinatorial complexity measure, and Eqn. (0.3) is the shrinking cell condition.

Consistency Results for Neural Network Classifier

Theorem (Lugosi & Zeger, 1995, IEEE Information Theory)

Consider a neural network with one hidden layer with bounded output weight having k hidden neurons and let σ be a logistic squasher. Let $F_{n,k}$ be the class of neural networks defined as

$$F_{n,k} = \left\{ \sum_{i=1}^{k} c_i \sigma(a_i^T z + b_i) + c_0 : k \in \mathbb{N}, a_i \in \mathbb{R}^{dm}, b_i, c_i \in \mathbb{R}, \sum_{i=0}^{k} |c_i| \leq \beta_n \right\}$$

and let ψ_n be the function that minimizes the empirical L_1 error over $\psi_n \in F_{n,k}$. It can be shown that if k and β_n satisfy

$$k \to \infty, \quad \beta_n \to \infty, \quad \frac{k\beta_n^2 log(k\beta_n)}{n} \to 0$$

then the classification rule

$$g_n(z) = \begin{cases} 0, & \text{if } \psi_n(z) \leq 1/2. \\ 1, & \text{otherwise.} \end{cases} \tag{0.4}$$

is universally consistent.

For universal convergence, the class over which the minimization is performed has to be defined carefully. Above theorem shows that this may be achieved by neural networks with k nodes, in which the range of output weights $c_0, c_1, \ldots, c_k$ is restricted.

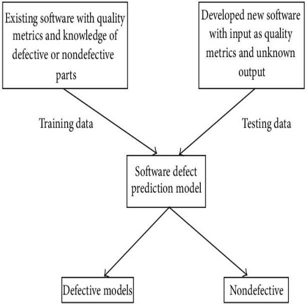
Part III: Imbalanced Classification Problem in Software Defect Prediction

Related Publication:

Tanujit Chakraborty and Ashis Kumar Chakraborty. “Hellinger Net: A Hybrid Imbalance Learning Model to Improve Software Defect Prediction”. **IEEE Transactions on Reliability** (2020). **(Read Online)**

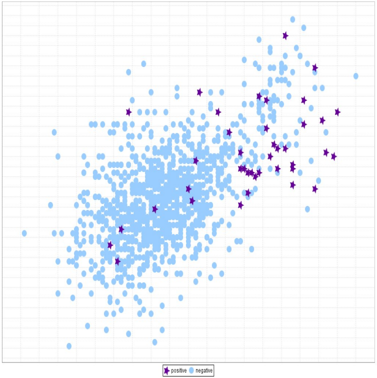
Motivation

- Software defect prediction is important to identify defects in the early phases of software development life cycle.
- This early identification and thereby removal of software defects is crucial to yield a cost-effective and good quality software product.
- Though, previous studies have successfully used machine learning techniques for software defect prediction, these techniques yield biased results when applied on imbalanced data sets.
- This study proposes an ensemble classifier, namely Hellinger Net, for software defect prediction on imbalanced NASA data sets.


Existing software with quality metrics and knowledge of defective or nondefective parts
Developed new software with input as quality metrics and unknown output
Training data
Testing data
Software defect prediction model
Defective models
Nondefective

Imbalanced Classification Problem

- Real-world data sets are usually skewed, in that many cases belong a larger class and fewer cases belong to a smaller yet usually more exciting class
- For example, consider a binary classification problem with the class distribution of 90 : 10. In this case, a straightforward method of guessing all instances to be positive class would achieve an accuracy of 90%.
- Learning from an imbalanced data set presents a tricky problem in which traditional learning algorithms perform poorly.
- Traditional classifiers usually aim to optimize the overall accuracy without considering the relative distribution of each class.



Sampling Techniques

- One way to deal with the imbalanced data problems is to modify the class distributions in the training data by applying sampling techniques to the data set
- Sampling technique either oversamples the minority class to match the size of the majority class or undersamples the majority class to match the size of the minority class.
- Synthetic minority oversampling technique (SMOTE) is among the most popular methods that oversamples the minority class by generating artificially interpolated data (Chawla et al., 2002, JAIR).
- TL (Tomek links) and ENN (edited nearest neighbor) are popular undersampling approaches (Batista et al., 2004, ACM SIGKDD).
- But these approaches have apparent deficiencies, such as undersampling majority instances may lose potentially useful information of the data set and oversampling increases the size of the training data set which may increase computational cost.
- To overcome these problems, "imbalanced data-oriented" algorithms are designed which can handle class imbalance without any modification to class distribution.

Effect of Class Imbalance on Decision Tree

Let X be attribute and Y be the response class. Here Y^+ denotes majority class, Y^- denotes minority class and n is the total number of instances. Also, let $X^{\geq} \longrightarrow Y^+$ and $X^{<} \longrightarrow Y^-$ be two rules generated by Classification Tree (CT). Table below shows the number of instances based on the rules created using CT.

Table: An example of notions of classification rules

class and attribute	$X^{\geq}$	$X^{<}$	sum of instances
Y^+	a	b	$a+b$
Y^-	c	d	$c+d$
sum of attributes	$a+c$	$b+d$	n

In the case of imbalanced data set the majority class is always much larger than the size of the minority class and thus we will always have $a+b >> c+d$. It is clear that the generation of rules based on confidence in CT is biased towards majority class.

Various measures, like information gain (IG), gini index (GI) and misclassification impurity (MI) expressed as a function of confidence, are used to decide which variable to split in the important feature selection stage, get affected by class imbalance.

Effect of Class Imbalance on Distance Measures

Table: An example of notions of classification rules

class and attribute	$X^{\geq}$	$X^{<}$	sum of instances
Y^{+}	a	b	$a+b$
Y^{-}	c	d	$c+d$
sum of attributes	$a+c$	$b+d$	n

Using Table, we compute the following:

$$P(Y^{+}/X^{\geq}) = \frac{a}{a+c} = \text{Confidence}(X^{\geq} \longrightarrow Y^{+})$$

For an imbalanced data set, Y^{+} will occur more frequently with $X^{\geq}$ & $X^{<}$ than to Y^{-}. So the concept of confidence is a fatal error in an imbalanced classification problem.

Entropy at node t is defined as:

$$\text{Entropy}(t) = -\sum_{j=1,2} P(j/t)log\big(P(j/t)\big)$$

Effect of Class Imbalance on Distance Measures

In binary classification, information gain for splitting a node t is defined as:

$$\text{IG} = \text{Entropy}(t) - \sum_{i=1,2} \frac{n_i}{n}\text{Entropy}(i) \tag{0.5}$$

where i represents one of the sub-nodes after splitting (assuming we have two sub nodes only), n_i is the number of instances in sub-node i and n is the total number of instances. The objective of classification using CT is to maximize IG which reduces to:

$$\text{Maximize}\left\{ - \sum_{i=1,2} \frac{n_i}{n}\text{Entropy}(i) \right\} \tag{0.6}$$

The maximization problem in eqn. (0.6) reduces to:

$$\begin{aligned} \text{Maximize}\Bigg\{ & \frac{n_1}{n}\Big[P(Y^+/X^{\geq})log\big(P(Y^+/X^{\geq})\big) + P(Y^-/X^{\geq})log\big(P(Y^-/X^{\geq})\big)] \\ & + \frac{n_2}{n}[P(Y^+/X^{<})log\big(P(Y^+/X^{<})\big) + P(Y^-/X^{<})log\big(P(Y^-/X^{<})\big)\Big]\Bigg\} \end{aligned} \tag{0.7}$$

The task of selecting the "best" set of features for node i are carried out by picking up the feature with maximum IG. As $P(Y^+/X^{\geq}) >> P(Y^-/X^{\geq})$, we face a problem while maximizing Eqn. (0.7).

Hellinger Distance

Let (Θ, λ) denote a measurable space. Let us suppose that P and Q be two continuous distributions with respect to the parameter λ having the densities p and q in a continuous space Ω, respectively. Define HD as follows:

$$d_H(P,Q) = \sqrt{\int_\Omega (\sqrt{p} - \sqrt{q})^2 d\lambda} = \sqrt{2\left(1 - \int_\Omega \sqrt{pq} d\lambda\right)}$$

where $\int_\Omega \sqrt{pq} d\lambda$ is the Hellinger integral. It is noted that HD doesn't depend on the choice of the parameter λ.

For the application of HD as a decision tree criterion, the final formulation can be written as follows:

$$HD = d_H(X_+, X_-) = \sqrt{\sum_{j=1}^{k} \left(\sqrt{\frac{|X_{+j}|}{|X_+|}} - \sqrt{\frac{|X_{-j}|}{|X_-|}} \right)^2}, \qquad (0.8)$$

where $|X_+|$ indicates the number of examples that belong to the majority class in training set and $|X_{+j}|$ is the subset of training set with the majority class and the value j for the feature X. The bigger the value of HD, the better is the discrimination between the features (Hellinger Distance Decision Tree, Chawla et al. 2008, ECML).

Hellinger Net : Basic idea

- Hellinger Net is composed of three basic steps:

  (a) Converting a DT into rules (HD is used as criterion);
  (b) Constructing a two hidden layered NN from the rules;
  (c) Training the MLP using gradient descent backpropagation (Rumelhart, Hinton (1988).

- In decision trees, the overfitting occurs when the size of the tree is too large compared to the number of training data.

- Instead of using pruning methods (removing child nodes), HN employs a backpropagation NN to give weights to nodes according to their significance.

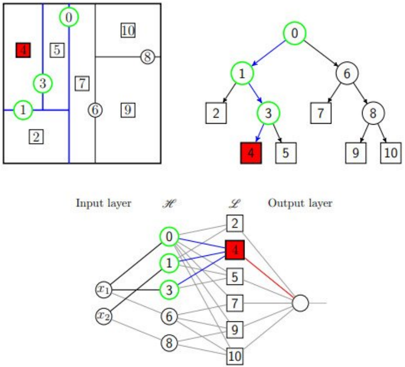


Fig: Graphical Representation of Hellinger Nets

Hellinger Net Algorithm

- Build a HDDT with $(k_n - 1)$ split nodes and k_n leaf nodes. HDDT is mapped into a two hidden layered MLP model having $(k_n - 1)$ and k_n hidden neurons in first hidden layer $(HL1)$ and second hidden layer $(HL2)$, respectively.

- The first hidden layer is called the partitioning layer which partitions the input feature spaces into different regions. It corresponds to the internal nodes of the DT. In $HL1$, the neurons compute all the tree split decisions and indicate the split directions for the inputs.

- Further, $HL1$ passes the information to $HL2$. The neurons in the second hidden layer represent the terminal nodes of the DT.

- The final layer is the output class label of the tree. Train the tree structured neural network using gradient descent backpropagation algorithm.

- Hellinger Net uses sigmoidal activation function instead of the relay-type activation function $\tau(u)$ with a hyperbolic tangent activation function $\sigma(u) = \tanh(u)$ which has a chosen range from -1 to 1.

- More precisely, the model uses $\sigma_1(u) = \sigma(\beta_1 u)$ at every neuron of the first hidden layer for better generalization, where β_1 is a positive hyper-parameter that determines the contrast of the hyperbolic tangent activation function.

Merits

- **Merits**:

  1. The additional training using backpropagation potentially improves the predictions of the HDDT and can deny tree pruning steps vis-a-vis the risk of overfitting;

  2. Hellinger Nets give weight to nodes according to their significance as determined by the gradient backpropagation algorithm;

  3. In Hellinger Nets, the neural network follows the built-in hierarchy of the originating tree since connections do not exist between all pairs of neurons in any two adjacent layers;

  4. Since the number of neurons in the hidden layers are fixed, thus the training time is less.

- **Theoretical developments**:

  1. Theoretical Consistency?

  2. Rate of Convergence?

On Theoretical Consistency

Theorem (Chakraborty et al., 2020, IEEE Transactions on Reliability)

Assume X is uniformly distributed in $[0,1]^p$ and $Y = \{0,1\}$. As $n \to \infty$ and for any $k_n, \beta_1, \beta_2 \to \infty$ if the following conditions are satisfied:

$$\begin{aligned}
&(A1) \quad \frac{k_n^4 log(\beta_2 k_n^4)}{n} \to 0, \\
&(A2) \quad \textit{there exists} \quad \delta > 0 \quad \textit{such that} \quad \frac{k_n^2}{n^{1-\delta}} \to 0, \\
&(A3) \quad \frac{k_n^2}{e^{2\beta_2}} \to 0, \quad \textit{and} \\
&(A4) \quad \frac{k_n^3 \beta_2}{\beta_1} \to 0,
\end{aligned}$$

then Hellinger Nets classifier is consistent.

The above Theorem states that with certain restrictions imposed on the number k_n of terminal nodes and the parameters β_1, β_2 being properly regulated as functions of n, the empirical L_1 risk-minimization provides local consistency of the Hellinger Nets classifier.

Rate of Convergence

Theorem (Chakraborty et al., 2020, IEEE Transactions on Reliability)

Assume that X *is uniformly distributed in* $[0,1]^p$ *and* $Y = \{0,1\}$ *and a function* $m : C^p \to \{0,1\}$ *is a Lipschitz* $(\delta; C)$*-smooth for any* $\delta \in [0,1]$. *Let* m_n *be the estimate that minimizes empirical* L_1*-risk and the network activation function* σ_i *satisfies Lipschitz property. Then for any* $n \geq max\{\beta_2, 2^{p+1}L\}$, *we have*

$$E \int_{[0,1]^p} \left| m_n(X) - m(X) \right| \mu(dx) = O\left(\frac{log(n)^6}{n} \right)^{\frac{2\delta}{2\delta+2p}}$$

- The proof of the Theorem is using Complexity Regularization Principles.
- The model will be able to circumvent the so-called problem of "curse of dimensionality".
- In practice, the larger the value of k_n, β_1, and β_2, the better the model performance is.

Applications: Data Sets

Data Sets: The proposed model is evaluated using five publicly available data sets from the area of Software Defect Prediction (NASA Metrics Data Program) available at Promise Software Engineering repository (`http://promise.site.uottawa.ca/SERepository/datasets-page.html`).

Table: Characteristics of the data sets used in experimental evaluation

Data set	Classes	Objects (n)	Number of feature (p)	Number of reported defects	Number of non-defects
CM1	2	498	21	49	449
JM1	2	10885	21	2106	8779
KC1	2	2109	21	326	1783
KC2	2	522	21	105	415
PC1	2	1109	21	77	1032

Performance Evaluation

The performance evaluation measure used in our experimental analysis is based on the confusion matrix in Table 2. Area under the receiver operating characteristic curve (AUC) is a popular metric for evaluating performances of imbalanced data sets and higher the value of AUC, the better the classifier is. AUC $= \frac{\text{Sensitivity}+\text{Specificity}}{2}$; where, Sensitivity $= \frac{TP}{TP+FN}$; Specificity $= \frac{TN}{FP+TN}$.

Table: Average AUC value for balanced data sets (using SMOTE and SMOTE+ENN) on different classifiers

Data	Sampling Techniques	kNN	CT	RF	ANN (with 1HL)	ANN (with 2HL)	RBFN
CM1	SMOTE	0.700	0.665	**0.722**	0.605	0.680	0.704
	SMOTE+ENN	0.685	0.650	0.708	0.600	0.652	0.700
JM1	SMOTE	0.758	0.745	0.762	0.740	0.735	0.764
	SMOTE+ENN	0.760	**0.778**	0.770	0.750	0.720	0.765
KC1	SMOTE	0.783	0.845	0.859	0.765	0.798	0.905
	SMOTE+ENN	0.801	0.850	0.875	0.798	0.807	**0.914**
KC2	SMOTE	0.927	0.965	**0.967**	0.933	0.942	0.954
	SMOTE+ENN	0.935	0.952	0.966	0.925	0.937	0.949
PC1	SMOTE	0.770	0.758	0.753	0.698	0.719	0.745
	SMOTE+ENN	**0.788**	0.760	0.761	0.712	0.725	0.748

Performance Evaluation

Highest AUC value in both the tables are highlighted with dark black for all the data sets. It is clear from computational experiments that our model stands as very much competitive with the current state-of-the-art models.

Table: AUC results (and their standard deviation) of classification algorithms over original imbalanced test data sets

Classifiers	CM1	JM1	KC1	KC2	PC1
CT	0.603 (0.04)	0.665 (0.03)	0.810 (0.04)	0.950 (0.00)	0.724 (0.02)
RF	0.690 (0.06)	0.725 (0.03)	0.850 (0.04)	0.964 (0.00)	0.747 (0.04)
k-NN	0.651 (0.03)	0.727 (0.01)	0.750 (0.03)	0.902 (0.02)	0.730 (0.05)
RBFN	0.652 (0.06)	0.723 (0.04)	0.884 (0.05)	0.935 (0.01)	0.725 (0.04)
HDDT	0.625 (0.04)	0.738 (0.04)	0.933 (0.02)	0.974 (0.00)	0.760 (0.02)
HDRF	0.636 (0.04)	0.742 (0.03)	0.939 (0.02)	**0.988** (0.00)	0.760 (0.03)
CCPDT	0.618 (0.05)	0.712 (0.05)	0.912 (0.03)	0.971 (0.00)	0.753 (0.01)
ANN (with 1HL)	0.585 (0.03)	0.700 (0.03)	0.768 (0.05)	0.918 (0.02)	0.649 (0.03)
ANN (with 2HL)	0.621 (0.02)	0.715 (0.02)	0.820 (0.04)	0.925 (0.01)	0.710 (0.03)
Hellinger Net	**0.720** (0.06)	**0.798** (0.04)	**0.964** (0.01)	0.985 (0.00)	**0.789** (0.05)

Simulation Study

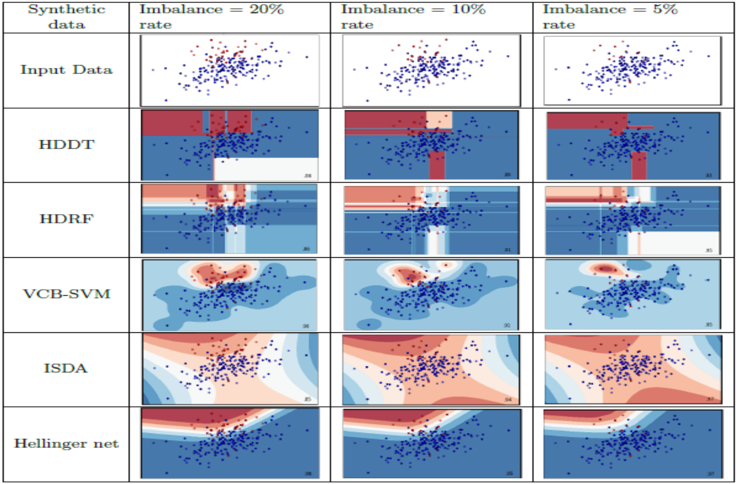
Simulated Data Sets: Three toy data sets (binary) are generated with weights = [0.2, 0.8], [0.1, 0.9] and [0.05, 0.95], i.e., data sets with imbalance rates of 20%, 10% and 5%, respectively. We added Gaussian noise to the data with the standard deviation equals to 0.5. This test problem is suitable for algorithms that can learn data imbalance problems in complex nonlinear manifolds.

Table: AUC results of different imbalanced classifiers on three synthetic data sets.

Imbalanced Classifiers	Simulated Data with IR = 20%	Simulated Data with IR = 10%	Simulated Data with IR = 5%
HDDT	0.80	0.85	0.91
HDRF	0.82	0.88	0.91
VCB-SVM	**0.87**	0.89	0.93
ISDA	0.84	0.91	0.90
Hellinger net	0.86	**0.92**	**0.95**

Simulation Study

A comparison of several imbalanced classifiers on synthetic data sets. The plots show training points in solid colors and testing points semi-transparent. The lower right in each plots shows the classification accuracy on the test set.



Conclusions

- Learning from an imbalanced data set presents a tricky problem in which traditional learning models perform poorly.
- Simply allocating half of the training examples to the minority class does not provide the optimal solution in most of the real-life problems.
- If one would like to work with the original data without taking recourse to sampling, our proposed hybrid methodology will be quite handy.
- We proposed ‘Hellinger Nets’, a hybrid learner, that first construct a tree and then simulate it using neural networks.
- We have proved the consistency of Hellinger Net model.

Crisis and Future Challenges

- The arena of research in learning from imbalanced data" continues to grow, largely driven by challenging problems including land cover classification, fraud detection, face recognition, spam and anomaly detection, medical diagnosis, etc

- The overarching question is "how to push the boundaries of prediction on the underrepresented or minority classes while managing the trade-off with false positives?"

- The usefulness and success of Random Forests and Deep Learning methods are evident. Can they be combined together to create a Deep Forest model that can deal with data imbalance problem?

- Use of Wasserstein Distance is of much use in the Machine Learning community for the last few decades. Some modification to the Wasserstein distance can be done and incorporated in the DT, RF, and Hellinger net model. This may improve the existing HDDT, HDRF and Hellinger Net models for imbalanced pattern classification.

Generalized Wasserstein Distance

Given two CDFs F_1 and F_2 on $\mathbb{R}$, let F denote the set of all joint distribution on $\mathbb{R}^2$ having F_1 and F_2 as marginals.

Definition

Given two CDFs F_1 and F_2 on $\mathbb{R}$, the Wasserstein distance between them is defined by

$$W_1(F_1, F_2) = \left[\inf_{F \in \Gamma(F_1, F_2)} \int_{\mathbb{R}^2} |x - y| dF(x, y) \right] \tag{0.9}$$

Proposed Generalized Wasserstein metric is a linear combination of the Wasserstein distance metric for discrete probability measures (d_E) and the absolute value of the differences in norms parameterized by a real number μ, and is defined as

$$D_\mu(x, y) = d_E\left(\frac{x}{\|x\|}, \frac{y}{\|y\|}\right) + \mu \left| \|x\| - \|y\| \right|$$

where, $d_E\left(\frac{x}{\|x\|}, \frac{y}{\|y\|}\right) = \sum_{i=1}^{p} \left| \left(\frac{x_i}{\|x\|} - \frac{y_i}{\|y\|} \right) \right|$. By using the actual norm information in the WD; we hope that the proposed GWD can deal data imbalance and outliers in the data sets better than Hellinger distance and others.

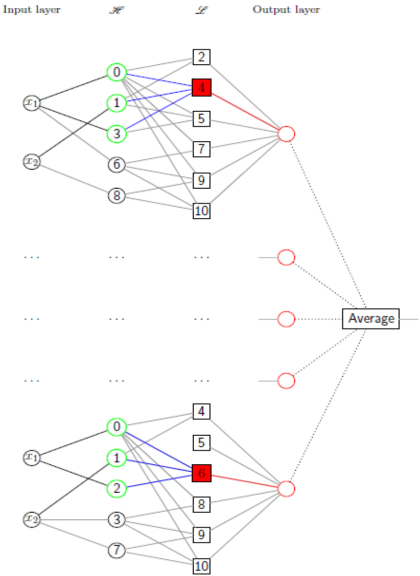
Generalized Wasserstein Distance

- It is straightforward to see that the proposed generalized Wasserstein distance is a metric.
- When x and y with well distinct norm are far away, we get large μ; and for small μ their renormalized versions only matters.
- By using a renormalized version of x and y and adding the norm information with weight parameters, we shall be able to make the WD metric skew-insensitive and useful to handle noisy data in imbalanced SDP problems.
- Generalized Wasserstein metric will be able to handle the highly imbalanced data problem within the Deep Forest framework.

Deep Forest - Individual Training

Finally, the project aims to make the proposed GWDF model:

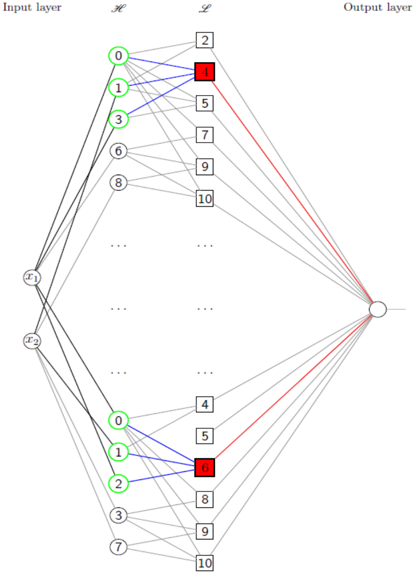
- scalable (the size of the data does not pose a problem),
- robust (work well in a wide variety of problems in the presence of noisy samples),
- accurate (achieve higher predictive accuracy),
- statistically sound (have desired asymptotic properties),
- easily interpretable for its effective implementation in land cover, aerial imagery and physiological data classification.



Deep Forest - Joint Training

We strongly desire that, whilst achieving competitive performance on imbalanced datasets in imbalanced data classification, GWDF would benefit from

- lightweight inference via conditional computation (sparse connected networks),
- hierarchical separation of features useful to the imbalanced learning task with generalized WD metric as tree splitting criteria,
- a mechanism to adapt the architecture to the size and complexity of the training dataset,



Generalized Wasserstein Deep Forest: Deliverables

- Generalized Wasserstein Deep Forest is a form of random forests enhanced with deep learned representations.
- Many existing tree-structured models are instantiations of the proposed GWDF model.
- The outcome of this work will be a suite of novel, principled, and interpretable deep learning techniques that would solve the imbalanced problem in SDP and others.
- We shall further investigate statistical consistency and rate of convergence for theoretical robustness of the proposed Wasserstein Deep Forest.
- Apart from the theoretical and computational development of the GWDF model and its implementation on real-world datasets, we aim to develop an implementation tool (a Toolbox in Python) for public use.

Part IV: Regression Estimation Problem in Process Efficiency Improvement

Related Publications:

Tanujit Chakraborty, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. "Radial basis neural tree model for improving waste recovery process in a paper industry", **Applied Stochastic Models in Business and Industry**, 36 (2020): 49-61. **(Read Online)**

Motivation

- This work is motivated by a particular problem in a modern paper manufacturing industry, in which maximum efficiency of the process fiber-filler recovery equipment, also known as Krofta supracell, is desired.

- As a by-product of the paper manufacturing process, a lot of unwanted materials along with valuable fibers and fillers come out as waste materials.

- The job of an efficient Krofta supracell is to separate the unwanted materials from the valuable ones so that fibers and fillers can be reused in the manufacturing process.

Fig: Krofta supracell

Process Efficiency Improvement Problem

- The Krofta recovery percentage was around 75%. The paper manufacturing company wants to improve the recovery percentage to 90%.
- To identify the important parameters affecting the Krofta efficiency, a failure mode and effect analysis (FMEA) was performed with the help of process experts.
- **Goal**: We would like to come up with a model that can help the manufacturing process industry to achieve an efficiency level of about 90% from the existing level of about 75% to improve the Krofta supracell recovery percentage.

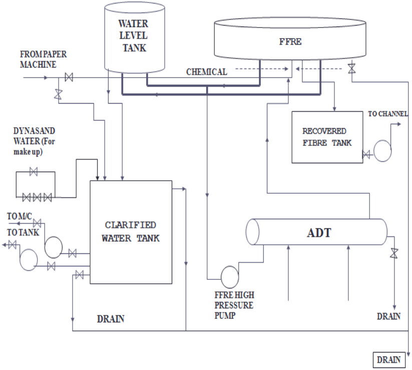


Fig: Process Flow Diagram of Krofta supracell

Process Data Set

- The data set collected for a year from the process on the following causal variables: Inlet Flow, Water Pressure (water inlet pressure to ADT), Air Pressure, Pressure of Air-Left, Pressure of Air-Right, Pressure of ADT-D Left, Pressure of ADT-D Right and Amount of chemical lubricants.
- The response variable (FFRE recovery percentage) lies between 20 to 100.
- This data set will be used for finding crucial process parameters and also finding a prediction model that can help the company for forecasting future recovery percentage of FFRE.

Table: Sample data set

Inlet Flow	Water Pressure	Air Pressure	Air-Left	Air-Right	ADT-D	ADT-D	Amount of	Recovery
						Left	Right	chemical
Percentage								
1448	6.4	5.8	1.0	2.1	3.2	4.0	2.0	96.80
1794	5.2	5.6	2.4	1.6	3.6	4.0	3.0	97.47
2995	6.0	6.0	1.5	4.5	4.0	4.8	4.0	28.87
1139	6.5	6.0	1.2	1.7	3.0	4.6	2.0	33.05
2899	6.2	5.7	2.0	1.2	3.1	4.0	2.0	97.91
1472	6.6	6.8	3.7	3.1	5.2	4.8	4.0	57.77
1703	6.2	6.0	2.9	1.0	3.0	4.2	2.0	26.94
1514	5.5	5.0	2.0	2.1	3.8	4.7	2.0	67.01
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Proposed Hybrid RBNT Model

- Apply RT algorithm to train and build a decision tree. Use the tree to extract the important features and find the splits between different adjacent values of the features.
- Choose the features that have minimum mean squared error as important input variables and record RT predicted outputs.
- Export important input variables along with an additional feature (prediction values of RT algorithm) to the RBFN model and a neural network is generated.
- RBFN model uses Gaussian kernel as an activation function, and parameter optimization is done using gradient descent algorithm. Finally, we obtain the final outputs.

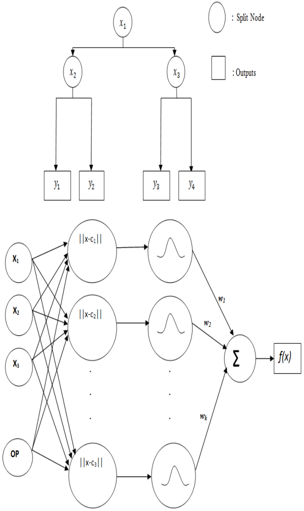


Fig: Flowchart of the Proposed Radial Basis Neural Tree Model

But...

- What will be the optimal Choice of the number of hidden nodes for the model? (Trial and Error!)
- Theoretical Consistency of the Model? (Statistical Learning Theory!)
- Importance of RT output in the second stage of the ensemble model? (Experimental or Theoretical Justification!)
- Experimental Evaluation and comparative study with single and hybrid ensemble models? (Important!)
- Can this model be useful for practitioner working in other disciplines but on similar types of problems? (Very Important!)

Improved Version of the Proposed Model

- First, apply the RT algorithm to train and build a decision tree and record important features.
- Using important input variables obtained from RT along with an additional input variable (RT output), a RBFN model (with one hidden layer) is generated.
- The optimum number of neurons in the hidden layer of the model to be chosen as $O(\sqrt{n/d_m log(n)})$ [to be discussed], where n, d_m are number of training samples and number of input features in RBFN model, respectively.

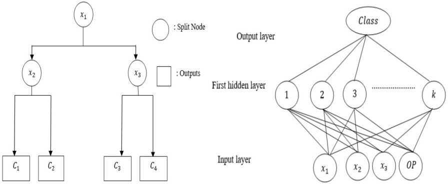


Figure: Graphical Presentation of the proposed ensemble model

Merits

- Can select important features from the data set;
- Suitable for Feature Selection cum Prediction Problems with limited data sets;
- Useful for high dimensional feature spaces in the data sets;
- Simple and Easily interpretable;
- “white-box-like” model, fast in implementation.

On Theoretical Consistency

Theorem (Chakraborty et al., 2020, Applied Stochastic Models)

Suppose $(\underline{X}, \underline{Y})$ *be a random vector in* $\mathbb{R}^p \times [-K, K]$ *and* L_n *be the training set of* n *outcomes of* $(\underline{X}, \underline{Y})$. *Finally if for every* n *and* $w_i \in \tilde{\Omega}_n$, *the induced subset* $(L_n)_{w_i}$ *contains at least* k_n *of the vectors of* $X_1, X_2, ..., X_n$, *then empirically optimal regression trees strategy employing axis parallel splits are consistent when the size* k_n *of the tree grows as* $o(\frac{n}{log(n)})$.

Theorem (Chakraborty et al., 2020, Applied Stochastic Models)

Consider a RBF network with Gaussian radial basis kernel having one hidden layer with $k\ (> 1)$ *nodes. If* $k \to \infty,\ b \to \infty$ *and* $\frac{kb^4 log(kb^2)}{n} \to 0$ *as* $n \to \infty$, *then RBFN model is said to be universally consistent for all distribution of* $(\underline{Z}, \underline{Y})$.

On the choice of Number of Hidden Neurons

- RBFN is a family of ANNs, consists of only a single hidden layer and uses radial basis function as an activation function, unlike feed forward neural network. RBF network with one hidden layer having k nodes for a fixed Gaussian function is given by the equation:

$$f(z_i) = \sum_{j=1}^{k} w_j \; exp\left(- \frac{\parallel z_i - c_i \parallel^2}{2\sigma_i^2} \right) + w_0,$$

  where $\sum_{j=0}^{k} |w_j| \leq b \; (> 0)$ and $c_1, c_2, ..., c_k \in \mathbb{R}^{d_m}$.

- For practical use, if the data set is limited, the recommendation is to use $k = \left(\sqrt{n/d_m log(n)}\right)$ for achieving utmost accuracy of the propose model.

Proposition (Chakraborty et al., 2019, Statistics & Probability Letters)

For any fixed d_m and training sequence ξ_n, let $Y \in [-K, K]$, and $m, f \in F_{n,k}$, if the neural network estimate m_n satisfies the above-mentioned regularity conditions of strong universal consistency and f satisfying $\int_{S_r} f^2(z)\mu(dz) < \infty$ where, S_r is a ball with radius r centered at 0, then the optimal choice of k is $O\left(\sqrt{\frac{n}{d_m log(n)}}\right)$.

Importance of RT output in neural net

- RT output also plays an important role in further modeling. It actually improves the performance of the model at a significant rate (can be shown using experimental results).
- We can use one hidden layer in Neural Network model due to the incorporation of RT output as an input information in ANN.
- RT predicted results provide some direction for the second stage modelling using ANN.
- Tree output estimates are probabilistic estimates, not from a direct mathematical or parametric model, thus direct correlationship with variables can't be estimated.
- It should be noted that one-hidden layer neural networks yield strong universal consistency and there is little theoretical gain in considering two or more hidden layered neural networks (Devroye, IEEE IT, 2013).
- To see the importance of RT given predicted results as a relevant feature, we introduced a non-linear measure of correlation between any feature and the actual values, namely C-correlation (Yu and Liu, 2004, JMLR), shown in (Chakraborty et al., 2019, Statistics & Probability Letters).

Experimental Evaluation

Popularly used performance metric are:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|;\ RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2};\ MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|;$$

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}\right];\ AdjR^2 = 1 - \left[\frac{(1-R^2)(n-1)}{n-d_m-1}\right];$$

where, $y_i, \overline{y}, \widehat{y}_i$ denote the actual value, average value and predicted value of the dependent variable, respectively for the i^{th} instant. Here n and d_m denote the number of data points and independent variables used for performance evaluation, respectively.

Table: Quantitative measure of performance for different regression models. Results are based on 10 fold cross validations. Mean values of the respective measures are reported with standard deviation within the bracket.

Models	**MAE**	**RMSE**	**MAPE**	R^2	**Adj(R^2)**
RT	11.691 (0.45)	16.927 (0.89)	29.010 (1.02)	59.028 (3.25)	55.304 (1.95)
ANN	12.334 (0.25)	17.073 (0.56)	27.564 (1.85)	58.310 (2.98)	54.529 (2.08)
SVR	12.460 (0.28)	20.362 (1.23)	40.010 (1.81)	40.174 (2.05)	35.325 (2.64)
BART	12.892 (0.59)	16.010 (1.25)	30.038 (1.95)	59.380 (2.50)	56.458 (1.75)
RBFN	13.926 (2.50)	18.757 (3.25)	32.48 (3.45)	49.689 (5.45)	46.335 (3.95)
Tsai Neural tree	10.895 (0.78)	16.012 (0.50)	24.021 (1.85)	65.120 (2.89)	62.946 (1.78)
Proposed Model	**9.226** (0.35)	**14.331** (0.82)	**20.187** (1.45)	**70.632** (2.00)	**68.675** (2.13)

Other Experiments on Process Data

Data Sets: The proposed model is evaluated using six publicly available from UCI Machine Learning repository (`https://archive.ics.uci.edu/ml/datasets.html`). These regression data sets have limited number of observations.

Table: Data set characteristics: number of samples and number of features, after removing observations with missing information or nonnumerical input features.

Sl. No.	**Data**	**Number of samples**	**Number of features**
1	Auto MPG	398	7
2	Concrete	1030	8
3	Forest Fires	517	10
4	Housing	506	13
5	Wisconsin	194	32

Table: Average RMSE results for each of the models across the different data sets

Data	**RT**	**ANN**	**SVR**	**BART**	**RBFN**	**Neural Tree**	**Our Model**
Auto MPG	3.950	4.260	5.720	3.220	4.595	3.300	**3.215**
Concrete	8.700	10.180	11.588	**5.540**	10.210	7.420	7.063
Forest Fires	75.138	90.702	91.985	65.890	82.804	**62.478**	64.411
Housing	4.980	9.054	12.520	3.978	7.871	4.590	**3.077**
Wisconsin	41.059	34.710	41.220	32.054	38.495	40.700	**23.659**

Application to Simulated data

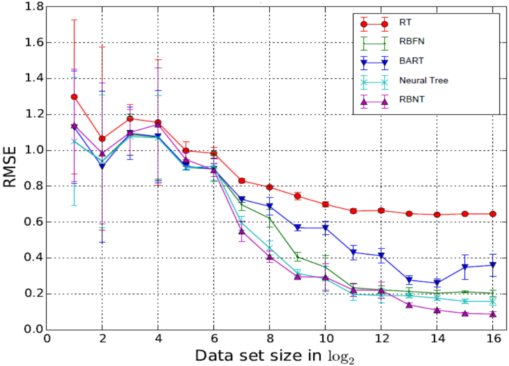
We investigate the asymptotic behavior of the proposed RBNT model on an artificial data set created by sampling inputs $\underline{x}$ uniformly from the p-dimensional hypercube $[0,1]^p$ and computing outputs y as

$$y(x) = \sum_{j=1}^{p} \sin\left(20x^{(j)} - 10\right) + \varepsilon,$$

where ε is a zero mean Gaussian noise with variance σ^2, which corrupts the deterministic signal. We choose $p = 2$ and $\sigma = 0.01$, and investigate the asymptotic behavior as the number of training samples increases. Figure in the next slide illustrates the RMSE for an increasing number of training samples and shows that the RBNT model error decreases much faster than other competitive model errors as sample size increases.

Asymptotic Behavior

This figure shows the test RMSE for synthetic data with exponentially increasing training set size (x-axis). Solid lines connect the mean RMSE values obtained across 3 randomly drawn data sets for each data set size, whereas error bars show the empirical standard deviation.



Conclusions

- In this chapter, we build a hybrid regression model for improving the process efficiency in a paper manufacturing company.
- Our study presented a hybrid RT-RBFN model that integrates RT and RBFN algorithm which gives more accuracy than all other competitive models to address the Krofta efficiency improvement problem.
- The proposed model is consistent, and when applied to other complex regression problems, it performed well as compared to other state-of-the-art.
- The usefulness and effectiveness of the model lie in its robustness and easy interpretability as compared to complex "black-box-like" models.

Part V: Conclusions and Future works

Conclusions & Future Work

- We developed some novel Hybrid Prediction models for various problems arising in classification and regressions.
- The problems arise from the area of Quality Control and Software Reliability.
- We studied several statistical properties of the proposed hybrid models.
- The scope of future research of the RBNT model will be to improve the model for survival data problems.
- Another scope of future research of the thesis will be to build Hybrid Models for Adversarial Machine Learning Problems.

References I

- Galton, Francis. Natural inheritance. Macmillan and Company, 1894.
- Fisher, Ronald A. “The precision of discriminant functions.” Annals of Eugenics 10.1 (1940): 422-429.
- Berkson, Joseph. “Application of the logistic function to bio-assay.” Journal of the American Statistical Association 39.227 (1944): 357-365.
- Fix, Evelyn, and Joseph L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. California Univ Berkeley, 1951.
- Parzen, Emanuel. “On estimation of a probability density function and mode.” The annals of mathematical statistics 33.3 (1962): 1065-1076.
- Breiman, Leo. Classification and regression trees. Routledge, 2017.
- Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. No. ICS-8506. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- Utgoff, Paul E. “Perceptron trees: A case study in hybrid concept representations.” Connection Science 1.4 (1989): 377-391.
- Friedman, Jerome H. “Multivariate adaptive regression splines.” The annals of statistics 19.1 (1991): 1-67.
- Cortes, Corinna, and Vladimir Vapnik. “Support-vector networks.” Machine learning 20.3 (1995): 273-297.
- Breiman, Leo. ”Random forests.” Machine learning 45.1 (2001): 5-32.
- Krizhevsky, A., I. Sutskever., and Hinton. G., “ImageNet Classification with Deep. Convolutional Neural Networks.” NIPS (2012).
- Goodfellow, Ian, et al. “Generative adversarial nets.” Advances in neural information processing systems. 2014.

References II

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” nature 521.7553 (2015): 436.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. “Deep bayesian active learning with image data.” Proceedings of the 34th International Conference on Machine Learning-Volume 70. 2017.

Chipman, Hugh A., Edward I. George, and Robert E. McCulloch. “BART: Bayesian additive regression trees.” The Annals of Applied Statistics 4.1 (2010): 266-298.

Lugosi, Gábor, and Andrew Nobel. “Consistency of data-driven histogram methods for density estimation and classification.” The Annals of Statistics 24.2 (1996): 687-706.

Nobel, Andrew. “Histogram regression estimation using data-dependent partitions.” The Annals of Statistics 24.3 (1996): 1084-1105.

Kearns, Michael J., and Yishay Mansour. “A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization.” ICML. Vol. 98. 1998.

Mansour, Yishay, and David A. McAllester. “Generalization Bounds for Decision Trees.” COLT. 2000.

Nobel, Andrew B. “Analysis of a complexity-based pruning scheme for classification trees.” IEEE Transactions on Information Theory 48.8 (2002): 2362-2368.

Denil, Misha, David Matheson, and Nando Freitas. “Consistency of online random forests.” International conference on machine learning. 2013.

Scornet, Erwan, Gérard Biau, and Jean-Philippe Vert. “Consistency of random forests.” The Annals of Statistics 43.4 (2015): 1716-1741.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators.” Neural networks 2.5 (1989): 359-366.

References III

Hinton, E. C., et al. “Neural representations of hunger and satiety in Prader–Willi syndrome.” International Journal of Obesity 30.2 (2006): 313.

Faragó, András, and Gábor Lugosi. “Strong universal consistency of neural network classifiers.” IEEE Transactions on Information Theory 39.4 (1993): 1146-1151.

Mhaskar, Hrushikesh Narhar. “Approximation properties of a multilayered feedforward artificial neural network.” Advances in Computational Mathematics 1.1 (1993): 61-80.

Hwang, JT Gene, and A. Adam Ding. “Prediction intervals for artificial neural networks.” Journal of the American Statistical Association 92.438 (1997): 748-757.

Hamers, Michael, and Michael Kohler. “Nonasymptotic bounds on the L 2 error of neural network regression estimates.” Annals of the Institute of Statistical Mathematics 58.1 (2006): 131-151.

Shaham, Uri, Alexander Cloninger, and Ronald R. Coifman. “Provable approximation properties for deep neural networks.” Applied and Computational Harmonic Analysis 44.3 (2018): 537-557.

Bauer, Benedikt, and Michael Kohler. “On deep learning as a remedy for the curse of dimensionality in nonparametric regression.” The Annals of Statistics 47.4 (2019): 2261-2285.

Lugosi, Gábor, and Kenneth Zeger. “Nonparametric estimation via empirical risk minimization.” IEEE Transactions on information theory 41.3 (1995): 677-687.

Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2004.

Sethi, Ishwar Krishnan. “Entropy nets: from decision trees to neural networks.” Proceedings of the IEEE 78.10 (1990): 1605-1613.

References IV

Sirat, J. A., and J. P. Nadal. “Neural trees: a new tool for classification.” Network: computation in neural systems 1.4 (1990): 423-438.

Jackson, Jeffrey C., and Mark Craven. “Learning sparse perceptrons.” Advances in Neural Information Processing Systems. 1996.

Bennett, Kristin P., and J. A. Blue. “A support vector machine approach to decision trees.” 1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence. Vol. 3. IEEE, 1998.

Jerez-Aragonés, José M., et al. “A combined neural network and decision trees model for prognosis of breast cancer relapse.” Artificial intelligence in medicine 27.1 (2003): 45-63.

Chen, Yuehui, Ajith Abraham, and Bo Yang. “Feature selection and classification using flexible neural tree.” Neurocomputing 70.1-3 (2006): 305-313.

Sugumaran, V., V. Muralidharan, and K. I. Ramachandran. “Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing.” Mechanical systems and signal processing 21.2 (2007): 930-942.

Nagi, Jawad, et al. “Convolutional neural support vector machines: hybrid visual pattern classifiers for multi-robot systems.” 2012 11th International Conference on Machine Learning and Applications. Vol. 1. IEEE, 2012.

Gjorgjevikj, Dejan, Gjorgji Madjarov, and SAŠO DŽEROSKI. “Hybrid decision tree architecture utilizing local svms for efficient multi-label learning.” International Journal of Pattern Recognition and Artificial Intelligence 27.07 (2013): 1351004.

Rota Bulo, Samuel, and Peter Kontschieder. “Neural decision forests for semantic image labelling.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

Kontschieder, Peter, et al. “Deep neural decision forests.” Proceedings of the IEEE international conference on computer vision. 2015.

Hinton, Geoffrey, and Nicholas Frosst. “Distilling a Neural Network Into a Soft Decision Tree.” (2017).

References V

- Yang, Yongxin, Irene Garcia Morillo, and Timothy M. Hospedales. "Deep neural decision trees." arXiv preprint arXiv:1806.06988 (2018).
- Peña-Ayala, Alejandro. "Educational data mining: A survey and a data mining-based analysis of recent works." Expert systems with applications 41.4 (2014): 1432-1462.
- Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.
- Batista, Gustavo EAPA, Ronaldo C. Prati, and Maria Carolina Monard. "A study of the behavior of several methods for balancing machine learning training data." ACM SIGKDD explorations newsletter 6.1 (2004): 20-29.
- Cieslak, David A., and Nitesh V. Chawla. "Learning decision trees for unbalanced data." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2008.
- Krzyzak, Adam, Tamás Linder, and C. Lugosi. "Nonparametric estimation and classification using radial basis function nets and empirical risk minimization." IEEE Transactions on Neural Networks 7.2 (1996): 475-487.
- Krzyzak, Adam, and Tamás Linder. "Radial basis function networks and complexity regularization in function learning." Advances in neural information processing systems. 1997.
- Cieslak, David A., et al. "Hellinger distance decision trees are robust and skew-insensitive." Data Mining and Knowledge Discovery 24.1 (2012): 136-158.
- Liu, Wei, et al. "A robust decision tree algorithm for imbalanced data sets." Proceedings of the 2010 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2010.
- Su, Chong, et al. "Improving random forest and rotation forest for highly imbalanced datasets." Intelligent Data Analysis 19.6 (2015): 1409-1432.

References VI

Daniels, Zachary Alan, and Dimitris N. Metaxas. “Addressing imbalance in multi-label classification using structured hellinger forests.” Thirty-First AAAI Conference on Artificial Intelligence. 2017.

Krofta, Milos. “Apparatus for clarification of water.” U.S. Patent No. 4,626,345. 2 Dec. 1986.

Tsai, Chia-Cheng, Mi-Cheng Lu, and Chih-Chiang Wei. “Decision tree–based classifier combined with neural-based predictor for water-stage forecasts in a river basin during typhoons: a case study in Taiwan.” Environmental engineering science 29.2 (2012): 108-116.

Drucker, Harris, et al. “Support vector regression machines.” Advances in neural information processing systems. 1997.

Box, George EP, and Gwilym M. Jenkins. “Time series analysis: Forecasting and control San Francisco.” Calif: Holden-Day (1976).

Faraway, Julian, and Chris Chatfield. “Time series forecasting with neural networks: a comparative study using the air line data.” Journal of the Royal Statistical Society: Series C (Applied Statistics) 47.2 (1998): 231-250.

Hyndman, Rob J., and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.

Tseng, Fang-Mei, Hsiao-Cheng Yu, and Gwo-Hsiung Tzeng. “Combining neural network model with seasonal time series ARIMA model.” Technological forecasting and social change 69.1 (2002): 71-87.

Zhang, G. Peter. “Time series forecasting using a hybrid ARIMA and neural network model.” Neurocomputing 50 (2003): 159-175.

Terui, Nobuhiko, and Herman K. Van Dijk. “Combined forecasts from linear and nonlinear time series models.” International Journal of Forecasting 18.3 (2002): 421-438.

Pai, Ping-Feng, and Chih-Sheng Lin. “A hybrid ARIMA and support vector machines model in stock price forecasting.” Omega 33.6 (2005): 497-505.

References VII

Yu, Lean, Shouyang Wang, and Kin Keung Lai. "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates." Computers & Operations Research 32.10 (2005): 2523-2541.

Huang, Shian-Chang. "Online option price forecasting by using unscented Kalman filters and support vector machines." Expert Systems with Applications 34.4 (2008): 2819-2825.

Aladag, Cagdas Hakan, Erol Egrioglu, and Cem Kadilar. "Forecasting nonlinear time series with a hybrid methodology." Applied Mathematics Letters 22.9 (2009): 1467-1470.

Khashei, Mehdi, and Mehdi Bijari. "An artificial neural network (p, d, q) model for timeseries forecasting." Expert Systems with applications 37.1 (2010): 479-489.

Faruk, Durdu Ömer. "A hybrid neural network and ARIMA model for water quality time series prediction." Engineering Applications of Artificial Intelligence 23.4 (2010): 586-594.

Chan, Kung S., and Howell Tong. "On the use of the deterministic Lyapunov function for the ergodicity of stochastic difference equations." Advances in applied probability 17.3 (1985): 666-678.

Khashei, Mehdi, and Mehdi Bijari. "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting." Applied Soft Computing 11.2 (2011): 2664-2675.

Chen, Kuan-Yu. "Combining linear and nonlinear model in forecasting tourism demand." Expert Systems with Applications 38.8 (2011): 10368-10376.

Khashei, Mehdi, and Mehdi Bijari. "A new class of hybrid models for time series forecasting." Expert Systems with Applications 39.4 (2012): 4344-4357.

Thank You