Predictive Clustering for Multi-Objective Regression

Sergei V. Gleyzer
DESY

Analysis Centre Seminar
14 February, 2013

Seminar Goal

Introduce and share a new technique:

Predictive clustering

- Developed outside of HEP
- Directly applicable to variety of problems in HEP
  - Multi-dimensional, multi-objective function estimation
    - Data are constantly multivariate (η, ϕ, E…)

Outline

- Introduction MVA methods
- Classification vs. Regression
- Single and Multi-Objective Regression
- Predictive Clustering Trees
- HEP Example Application
- Summary

Practicum

Download toy data and example

http://cern.ch/sergei/clusexample.tgz

unpack and try

Multivariate Methods

MVA Methods solve problems by building complex systems from underlying variables

Developed in Machine Learning (1980s)

Typical Applications:

Classification: Is this an apple or a pear?

Function Estimation: How many Dr.'s are present?

Forecasting: Who will be here at the end?

General Methodology

Machine-Learning view point: **Classification**

Distinguish f(x), **g(x)** using Training set of observations

{**inputs** , **outputs**}

Pass observations into a learning algorithm

neural network, decision tree

that produces **outputs** in response to **inputs**

Use another Testing set of observations to evaluate

Classification

Is this event a SUSY/Higgs event?

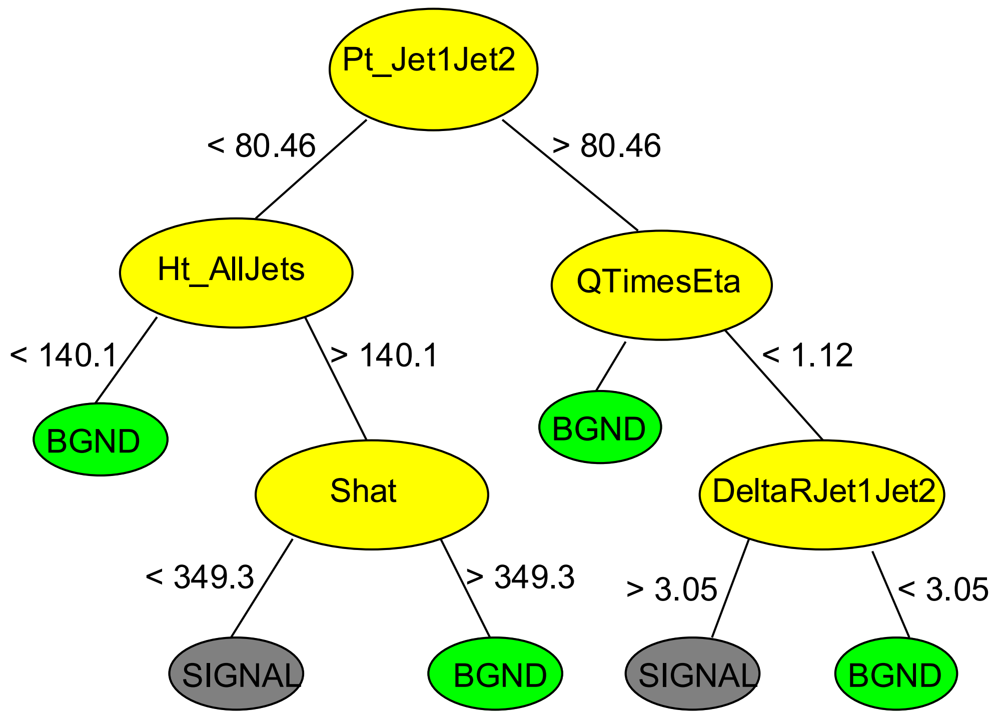
Plethora of methods:

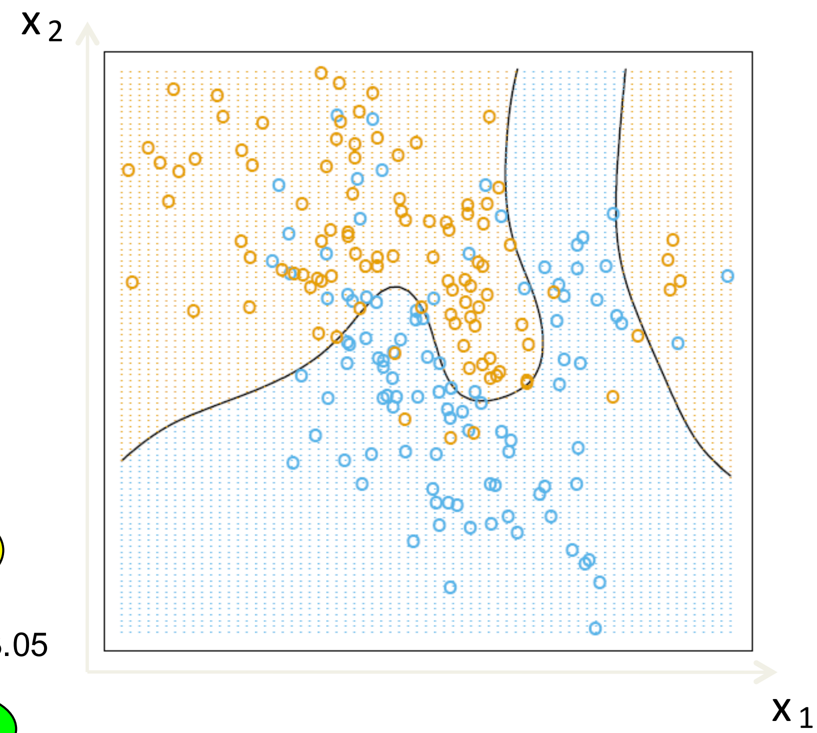
- **Neural Networks** ←
- **Boosted Decision Trees** ←
- **Support Vector Machines**
- **etc**

Describe in detail

Usually 5-30% improvement over **expert decisions**

Classification Example

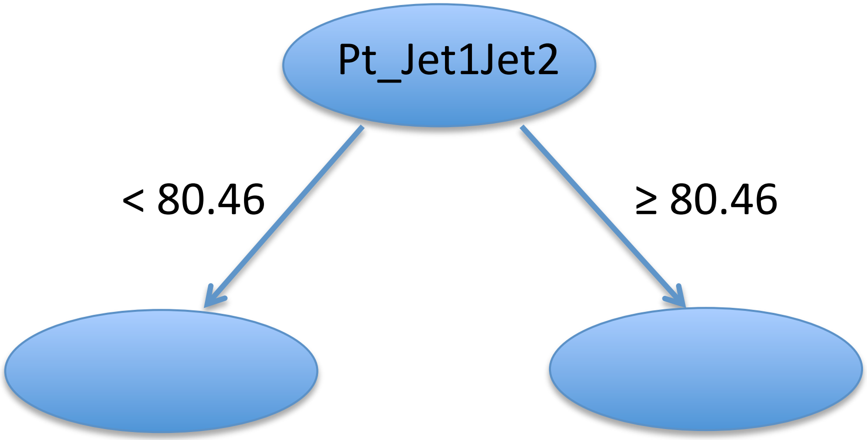




Decision Trees

Building a tree:

- Scan along each variable and propose a DECISION:
  - Cut on a variable value that maximizes class separation (branching into two)



Decision Optimization

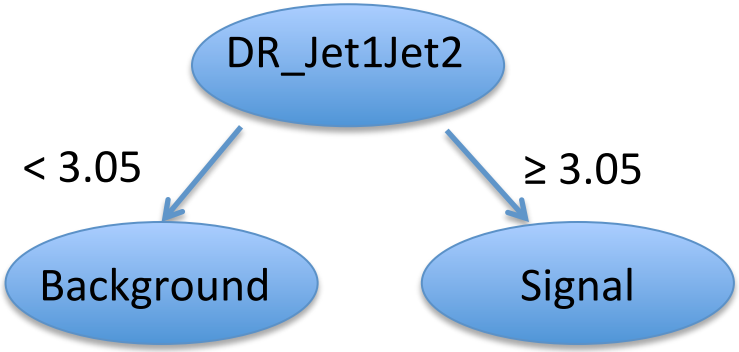
Compare decisions proposed by all variables at each juncture to select one optimal decision

- use information entropy to evaluate "information" gain from a proposed split
  - based on subsample purities (s/s+b)

- **"Greedy" algorithm**: each decision is irreversible and affects the next (very much like life)

Decision Trees

- **Stopping criteria:** no further improvement in separation from further branching
  - Sometimes maximum tree size is set a priori
  - Terminal leaf node is reached
  - Class assignment



Pruning

Decision trees can grow large and risk over-fitting the data

Improve tree by removing less powerful and possibly noisy parts: **Pruning**

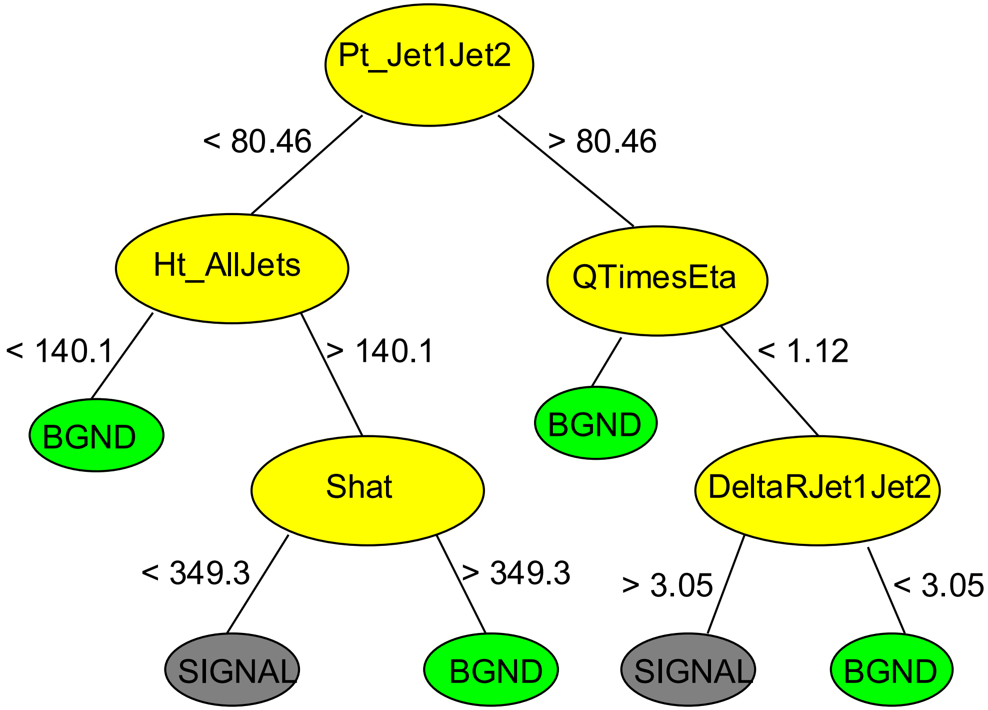
- Begin from the leaves and work back up
- Pruned trees smaller in size, more effective and easier to interpret

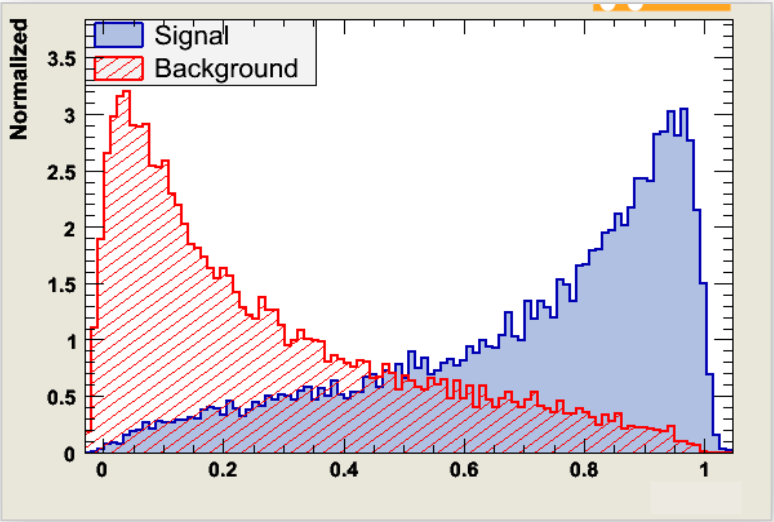
Boosting

Train in several stages:

- **Introduce event weights**
  - ADABoost: Freund & Schapire 1997
  - Misclassified events carry greater weight in subsequent training stages
  - Classify with a majority vote from all trees
- **Works very well to improve classification power of “greedy” decision trees**
  - sometimes used with other classifiers

Classification Example





Ensembles Methods

- General ensemble methods construct a set of classifiers for a given task
- Classify new instances by taking a vote on their predictions
- **Bagging:** combine trees grown from "re-sampled" training data with replacement
- **Random Forests:** use random subsets of training data and random variable sets for splitting
- **Rule Ensembles:** construct rules from trees

Rule Ensembles

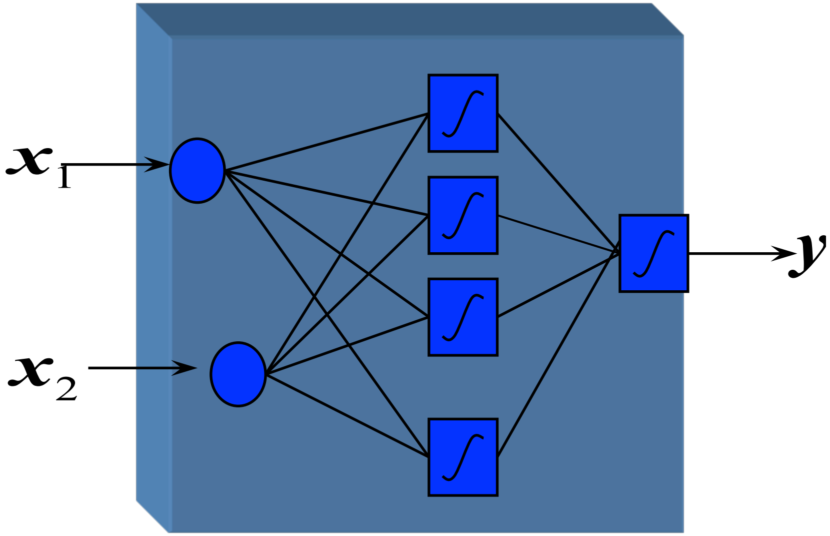
Decision trees can be transformed into a set of **{if, then… else}** rules

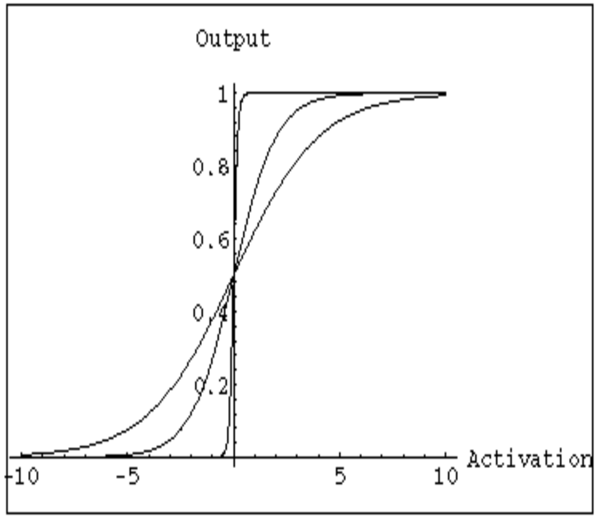
Start at the root and follow a unique path to a leaf

Simple rules form powerful classifiers in a weighted ensemble when assigning event classes based on majority decision

– Some rules slightly better than random guessing

Neural Networks



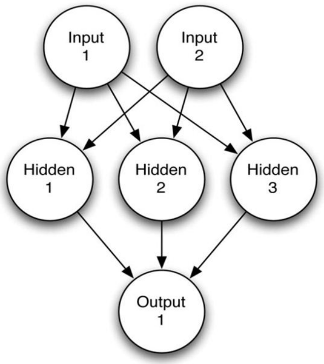


$$F = \sum_j \omega_{kj} f(\sum_i \omega_{ji} x_i + \theta_j) + \theta_k; \qquad y = \frac{1}{1 + e^{-F}}$$

Neural Networks-2

Compute optimal network weights with derivatives dE/dw

- Calculate gradients of errors for adjustable weights



Inputs go forward in feed-forward neural networks

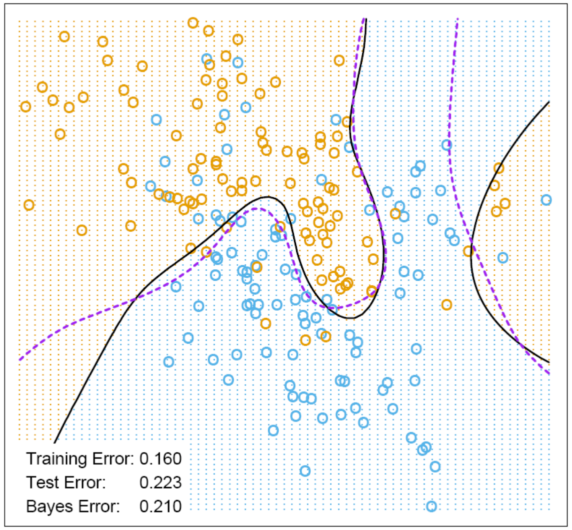
Errors go backward! **Backpropagation**

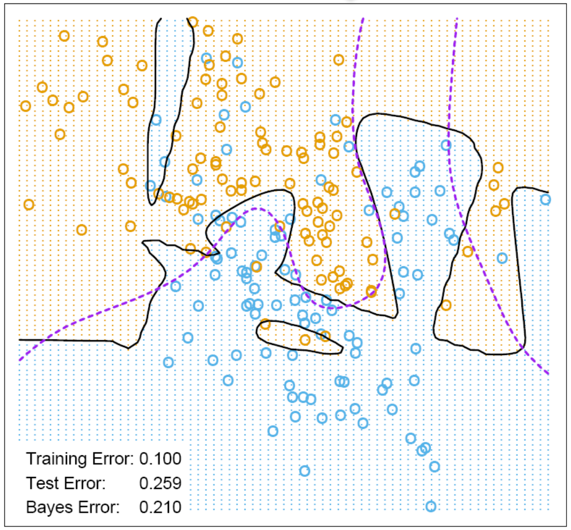
Neural Networks-3

Can approximate any continuous function

Complexity determined by number of hidden layers and hidden nodes/layer

Watch out for overtraining





Other Methods

Partial List of Classification Methods:

- Bayesian Neural Networks
- **Decision Trees ✓**
- Genetic Algorithms
- Linear Discriminants
- **Neural Networks ✓**
- Random Forests
- Random Grid Search
- **Rule Ensembles ✓**
- Support Vector Machines

✓ Discussed in this talk

Methodology II

Machine-Learning view point: **Function estimation**

Learn f(x) using a Training set of observations
{**inputs** , **outputs**}

feed observations into a learning algorithm
neural network, decision tree
that produces **outputs** in response to **inputs**

use another set of observations to evaluate

Function Estimation

Comet Problem by Gauss (1805): Approximate trajectory of a comet from observations

Approach: minimize difference between measurement and predictions in a systematic fashion

Vary regression model parameters

HEP Regression Example

Improve calorimeter resolution by applying regression

Inputs: electromagnetic shower information, other calorimetric variables

Target Output: calorimeter energy

Function estimation

- Think of decision tree as **multidimensional histogram**
  - Bins are recursively constructed
  - Each associated to the value of f(x) to be approximated

- To go from classification to regression change the evaluation criteria used in the learning algorithm
  - from **maximum separation gain** to **minimal variance** from resulting cuts

Extension: More Classes

Classification:

- Relatively easy to extend existing classifiers to handle more classes: just add more classes

Regression:

- Very hard to do well
- Nevertheless, very practical
- Less explored area in machine learning

1-Function Limitation

For problems that require simultaneous estimation of N functions (that are possibly related)

- N single-function regression model solution is too cumbersome
- Also less accurate
- Correlations among functions may be important and need to be accounted for

Multi-function regression models are a better solution in this case

Multi-Objective Models

- Properly take into account **dependencies** between output attributes (their correlations)

- **improved performance results** compared to single-objective models, especially in ensembles

- usually smaller and easier to interpret

- very useful for transformations

Predictive Clustering

Example of a **multi-function regression** model based on trees or rules

- **Decision trees** are equated to clustering trees by P. Langley in 1996, first noted by Fisher in 1993
- **Cluster "hierarchy"**

  Each tree node corresponds to a cluster

  Root node contains full dataset partitioned recursively into sub-clusters

Cluster Concept

Use **decision tree induction** to obtain clusters with:

- **minimal intra-cluster distance**
  - between examples from the same cluster
- **maximal inter-cluster distance**
  - between examples from different clusters
  - In classification trees distance metric is class enthropy

CLUS

Predictive clustering implementation

- Decision tree and rule induction system
- Designed for multi-task learning and multi-label classification
- Well-suited for both classification and regression problems

CLUS Example Setup

14 input variables {a, b, c, d…}
- 4 of them strongly correlated

14 target outputs to estimate {A, B, C, D…}
- 4 of them strongly correlated

Challenge: build a predictive model to describe simultaneously all the outputs {A,B,C,D…}, provided a corresponding set of inputs.

For example: These can be correlated EM shower-shapes

Procedure

Split data into disjoint Training and Testing Sets

- odd/even, randomize

Train the predictive clustering model by providing a “map” between inputs and outputs. Let it learn.

Evaluate: Use the Test set to compare predictions on “unseen” data to the Target values of the outputs.

Predictive Clustering Rules

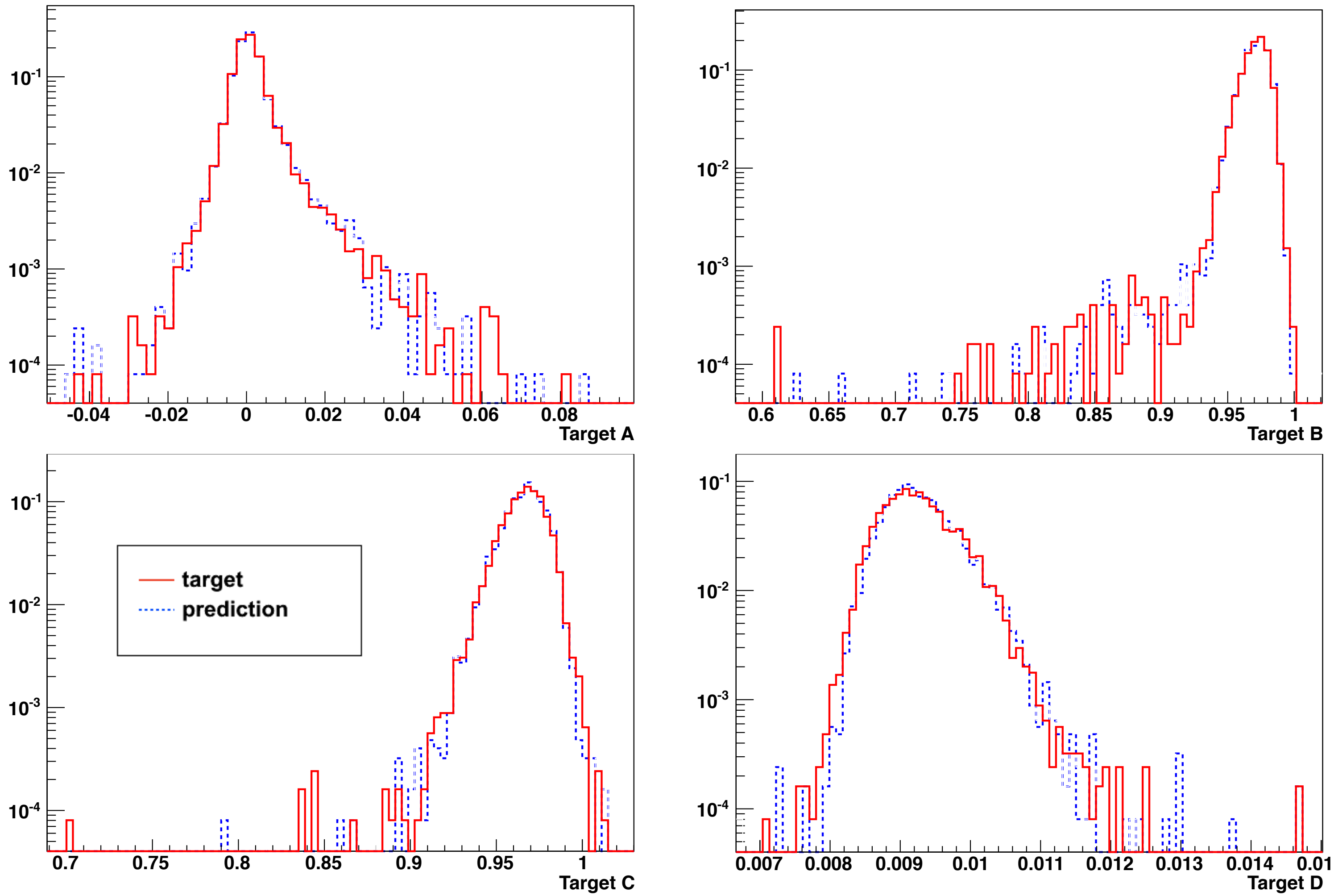
Predictive clustering rules can be constructed from predictive clustering trees

Main difference: simple rules focus on the accuracy connected to the target

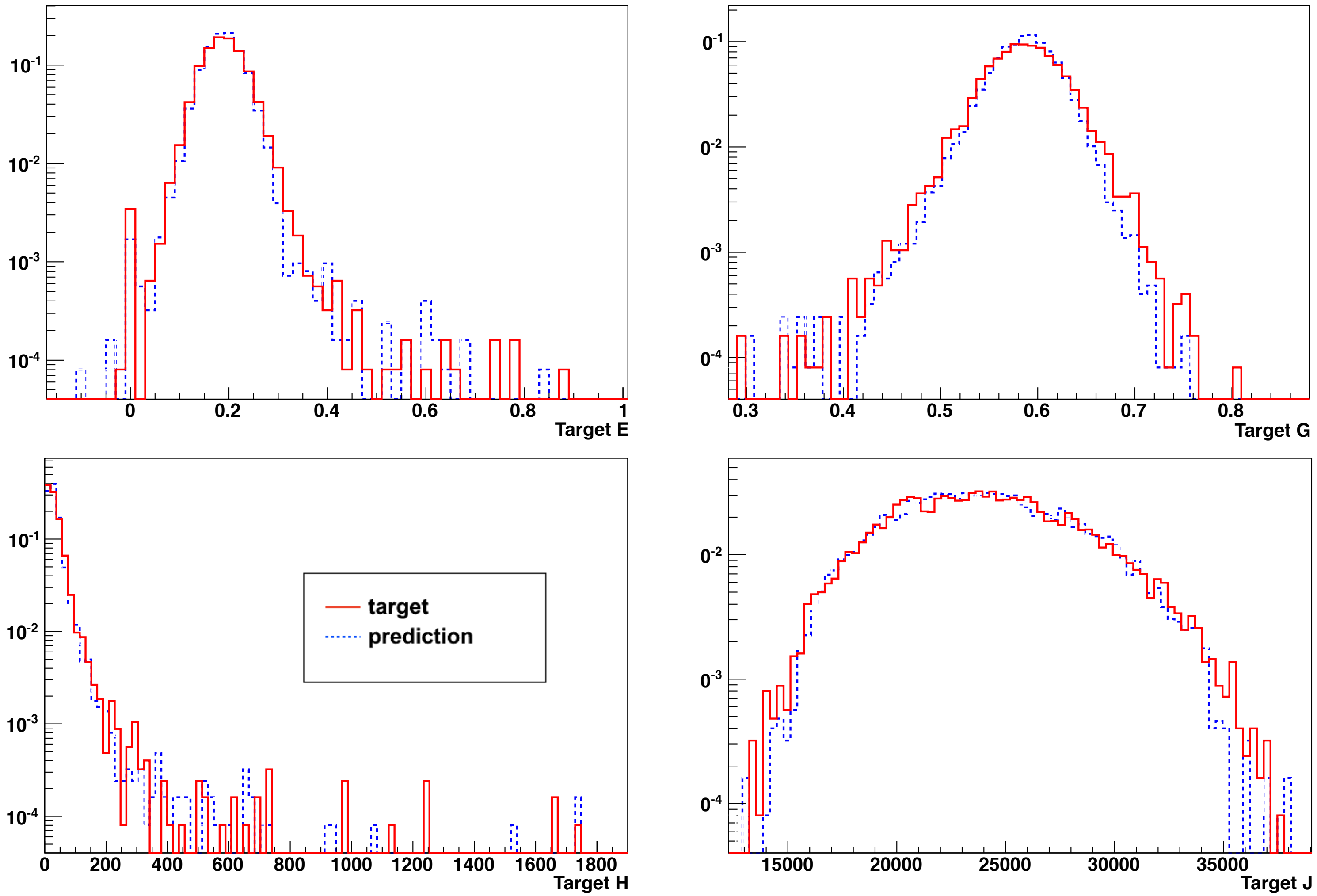
Predictive clustering rules focus on:

- target attribute accuracy
- tight or compact rule coverage of the instances by computing their distance metric

A Simple CLUS Example

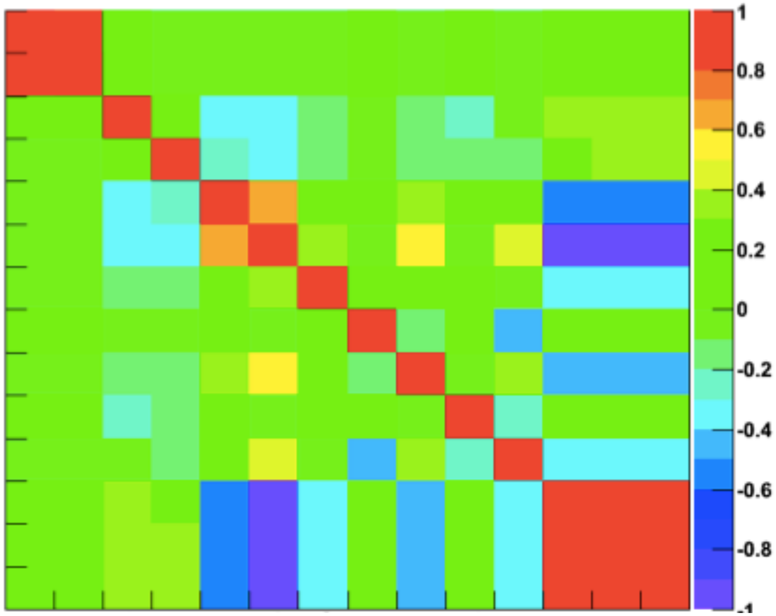


A Simple CLUS Example

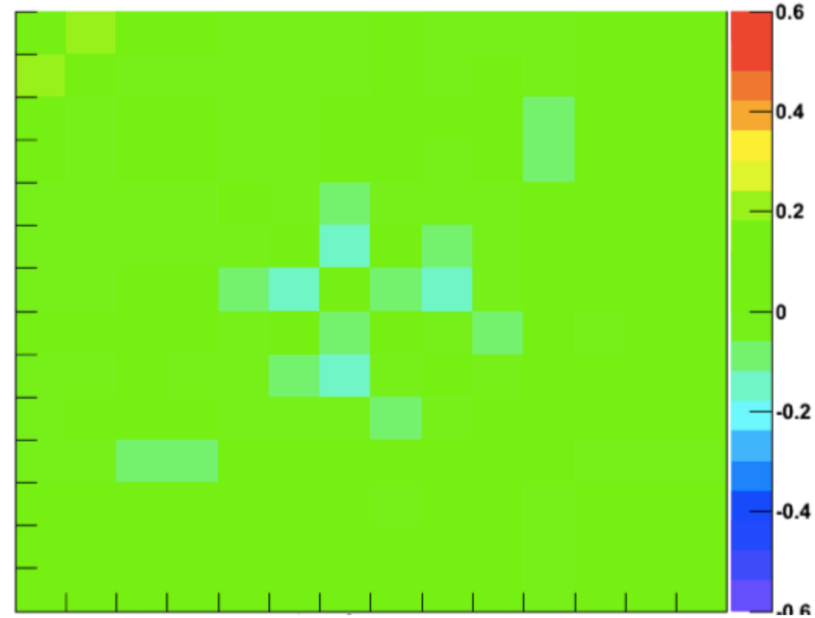


Correlations

Target Correlations



Prediction-Target Difference



Very close to Zero

Summary

- **Predictive clustering** is a robust method for simultaneous multi-function estimation
- **Functions are well reproduced** and correlations among variables preserved in the PCT model, good agreement with expected correlations
- **Ensemble methods** including bagging and rule ensembles are available for use with the CLUS package: try them ☺

Further Reading and Help

- Useful papers about CLUS:
  - **http://dtai.cs.kuleuven.be/clus/publications.html**

- CLUS Website:
  - **http://dtai.cs.kuleuven.be/clus/**
  - **http://dtai.cs.kuleuven.be/clus/hmcdatasets/ Toy data**

- **Local Experts @ DESY available for help and instructions:**
  - Myself (sergei.gleyzer@desy.de) and Chris Hengler (christopher.hengler@desy.de)

PHYSICS AT THE TERA SCALE
Helmholtz Alliance

DESY

The END
Happy Valentine's Day