StatPatternRecognition: A C++ Package for Multivariate Classification of HEP Data

Ilya Narsky, Caltech
presented by Julian Bunn, Caltech

Motivation and documentation

- Why write a new software package?
  - must be free
  - must be C++
  - must implement boosted decision trees and random forest
  - must be easily adjustable for optimization of HEP-specific figures of merit
- Docs:
  - links to SPR papers and talks on author's web page – http://www.hep.caltech.edu/~narsky/spr.html
  - detailed README included in the package

Classifiers in StatPatternRecognition

- binary split (aka “binary stump”, aka “cut”)
- linear and quadratic discriminant analysis
- decision trees
- bump hunting (PRIM, Friedman and Fisher)
- AdaBoost (Freund and Schapire)
- bagging and random forest (Breiman)
- variant of arc-x4 (Breiman)
- multi-class learner (Allwein, Schapire and Singer)
- combiner of classifiers
- interfaces to Stuttgart Neural Network Simulator methods:
  - feedforward backpropagation Neural Net
  - radial basis function Neural Net
  - These are not for training! Just for reading the saved net configuration and classifying new data.

Highlights: bump hunter

- Optimization of rectangular cuts (so beloved by many physicists) can be carried out using the bump hunter with an appropriate figure-of-merit
- A dozen of various FOM's have been implemented in the package:
  - signal significance $S/\sqrt{S+B}$, signal purity $S/(S+B)$, 90% Bayesian upper limit, potential for discovery $2(\sqrt{S+B}-\sqrt{B})$, Punzi's criterion $S/(0.5N_{\sigma}+\sqrt{B})$ etc
  - all plugged through an abstract interface

Highlights: boosted decision trees and random forest

- Two most powerful off-the-shelf **multivariate** classifiers
- Unlike neural nets, trees can easily deal with strongly correlated inputs, inputs of the mixed type (continuous + discrete), and missing values
- For decision trees, CPU scales linearly vs the number of dimensions, while for neural nets it scales quadratically or worse
- Physics analysis of new kind – use dozens or hundreds of input variables and let the multivariate classifier sort them out!

Other methods in StatPatternRecognition

- Bootstrap
  - tool for evaluating bias and variance of an estimator by resampling
  - see, e.g., Efron & Tibshirani, "An Introduction to the Bootstrap"
- Cross-validation
  - tool for choosing classifier parameters by splitting the data sample into subsets, training on one subset and validating on another
- Estimation of means, covariance matrix and kurtosis for a multivariate sample
- Goodness-of-fit evaluation using decision trees (Friedman, Phystat 2003)

Framework

- Object Oriented design
  - the package can be easily extended by supplying new implementations to existing interfaces
- User can impose selection criteria on input data
- User can work with multiple classes and combine them in two groups (signal and background) using easy-to-understand syntax
- User can save input data and classifier output in Ascii, Hbook or Root formats
- Input data can be read either from Ascii or Root
- SPR understands weighted data
- User can save classifier configuration into an Ascii file and resume training from the saved configuration
  - essential for training classifiers on huge datasets

Touch and feel

- No GUI support. User has a choice of
  - building supplied executables, one for each major method. All executables have a flexible set of command-line options.
  - writing his own C++ code. Examples are effectively provided in the aforementioned executables.

Example: training of random forest

```
➢ SprBaggerDecisionTreeApp -n 100 -y "1,3;2,4,5" -l 10 -s 6
-z "p,theta" -g 1 -f bagtree.spr -t validation.pat -d 5
training.pat
```

Build 100 decision trees with at least 10 events per leaf and save the trained classifier configuration into bagtree.spr. Display quadratic loss (-g 1) for validation data every 5 trees. Randomly select 6 variables (-s 6) to be considered for each decision split. Exclude variables “p” and “theta” from optimization. Group classes 1 and 3 into the background category and classes 2, 4, and 5 into the signal category.

External dependencies

- CLHEP (for matrix algebra)
- CERNLIB
  - random number generator for bootstrapping
  - probability of the tail of chisq distribution for evaluation of the consistency of a correlation coefficient with zero
  - CERNLIB dependency can be easily removed if alternative implementations are provided
- Root for input/output or Hbook for output only
  - optional since user can choose Ascii
- SPR was written for Unix in HEP environment
  - an incomplete adapted version of SPR for .NET is also available

Installation

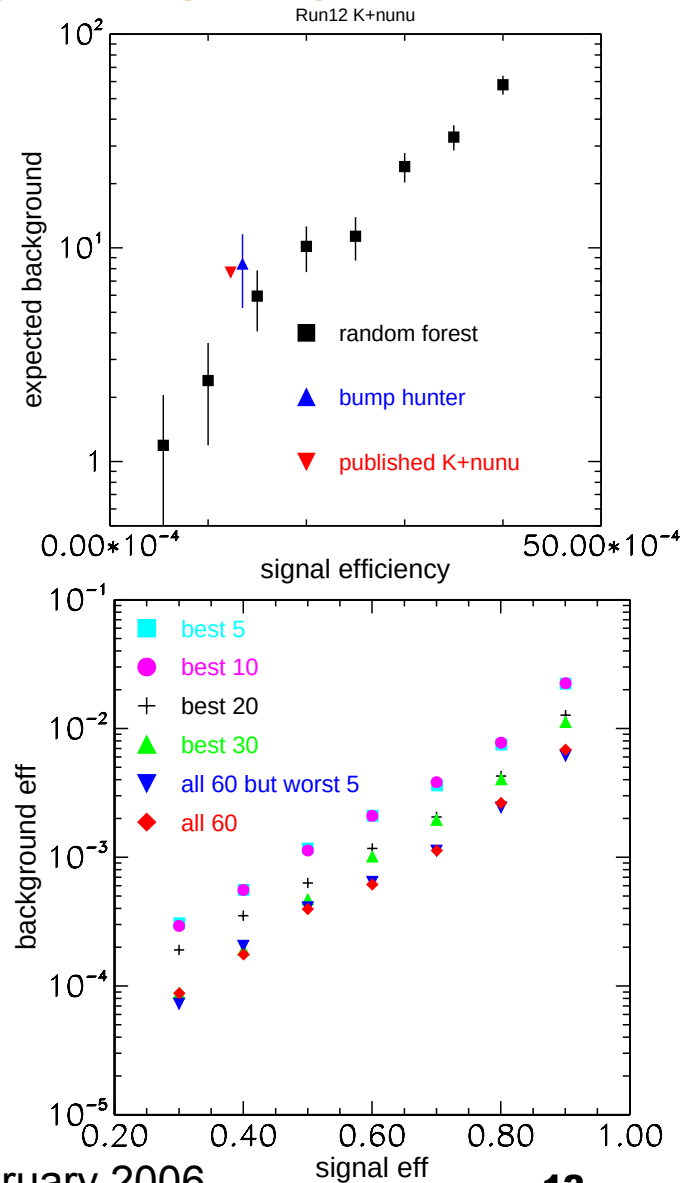
- In BaBar CVS – no installation for BaBar users
- Outside BaBar
  - send email to narsky@hep.caltech.edu for the latest tag of the package
  - write a Makefile that resolves references to CLHEP, CERNLIB and (optional) Root/Hbook
    - an example of Makefile can be found at http://www.hep.caltech.edu/~narsky/spr.html
  - replace BaBar-specific SprTupleWriter in the executables with SprAsciiWriter or SprRootWriter (both BaBar-independent)
  - You are ready to go!!!

Performance

- Tested on up to 1M events in up to 200 dimensions
- Applied to several physics datasets
  - $B^+ \to \gamma l^+ \nu$ and $B \to K^{(*)} \nu\nu$ analyses at BaBar
  - muon ID and B0/B0bar tagging at BaBar
  - two datasets from other experiments provided for testing only
- Benchmarks
  - application of random forest to 90k of 4D data showed a 10-fold reduction in CPU time compared to random forest implemented in R
  - comparable speed and similar performance (for boosted decision trees) to those of m-boost (Byron Roe's C implementation of boosted decision trees and random forest used for PID at MiniBoone)

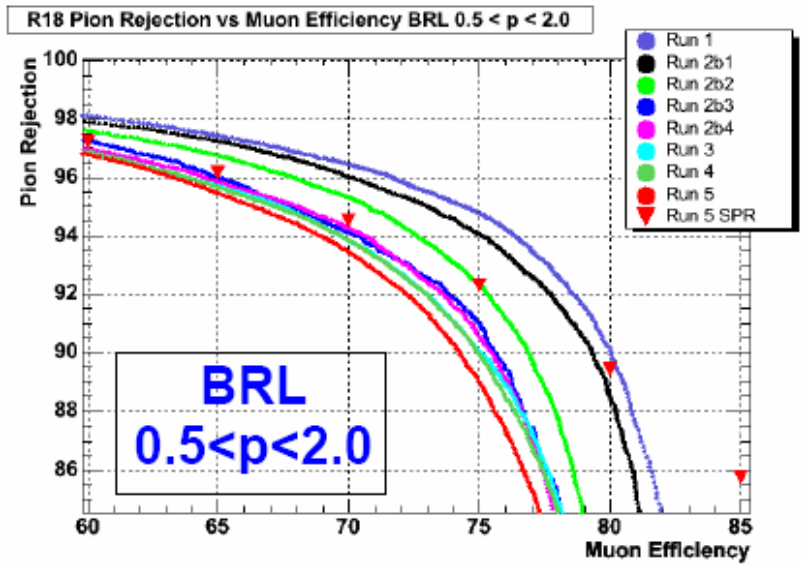
Search for $B^+ \to K^{(*)+} \nu\nu$ at BaBar

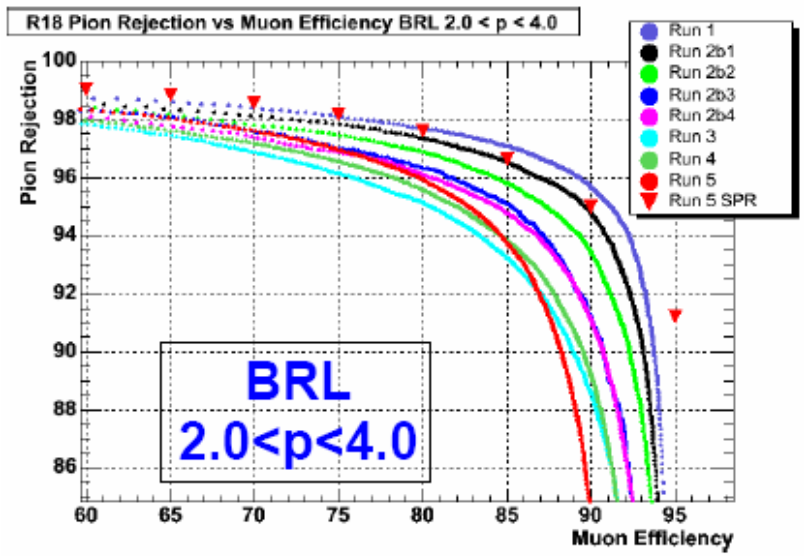
- **In 2004, BaBar published an upper limit on the rare decay $B^+ \to K^+ \nu\nu$ using ~90M BBbar pairs, based on rectangular cut optimization.**
- **We have recently tested SPR multivariate techniques on the same dataset (semileptonic tag only) using 60 input variables.**
- **Bump hunter (rectangular cuts):**
  - **signal efficiency and expected background are very close to those for published analysis**
  - **but different variables have been selected by the algorithm**
- **Random forest reduces background by a factor of ~3 for the same amount of signal.**
- **All 60 variables contribute to separation of signal, as shown in the bottom plot.**



Muon PID at BaBar

- To separate muons from pions, 17 variables are used from two subsystems:
  - Instrumented Flux Return
  - ElectroMagnetic Calorimeter
- At present BaBar uses a two-stage neural net:
  - NN for EMC variables only
  - an overall NN for IFR variables + output of EMC NN
- SPR random forest gives a substantial improvement in performance, especially in the barrel part of the detector: Compare red circles and red triangles.
- Note: Degradation of RPC's causes decrease in performance over time.





Acknowledgments

A number of people have contributed to this work, either by useful discussions or code submissions or by providing datasets for experimentation (listing in alphabetical order):

Josh Boehm (Harvard)
Maarten Bruinsma (UC Irvine)
Ed Chen (Caltech)
Gregory Dubois-Felsmann (Caltech)
Kevin Flood (Wisconsin)
Akram Khan (Brunel)
Michael Miller (MIT)
Frank Porter (Caltech)
Harrison Prosper (Florida)
Byron Roe (Michigan)
Mayly Sanchez (Harvard)
Gabriella Sciolla (MIT)
Jan Strube (Oregon)

Status

- **SPR is used in several (≥5) BaBar physics analyses.**
- **It is planned that SPR will be used for the new muon selector at BaBar.**
- **There are ~15 subscribers outside BaBar on the SPR mailing list.**
- **I am committed to support of the package. Please submit bug reports.**
- **However, in the near future I will be too busy to do any development.**
- **If you are (not a novice) C++ programmer and would like to contribute, please get in touch with me (narsky@hep.caltech.edu) to discuss options.**

Summary

- A C++ package for multivariate classification has been developed for HEP analysis.
- It has been tested on many idealistic and practical examples.
- In several case studies, SPR gives a significant improvement over the method used otherwise.
- SPR is used by dozens of people in the community. Would you like to join the list?