

# Alternative classifiers for B0/B0bar tagging

*Ilya Narsky, Caltech*

*Akram Khan, Brunel*

thanks to Maarten Bruinsma and Gabriella Sciolla

# Status

- StatPatternRecognition
  - a C++ package for multivariate classification
  - see my talk on Wednesday afternoon
- What has been done
  - B0/B0bar tagging samples have been used to test and optimize the performance of StatPatternRecognition
  - I worked irregularly on this project (not my top priority)
  - tested new classifiers on 9 subtaggers
  - looked at the possibility of B0/B0bar separation without subtagging (that is, try all interesting variables at once)
- Overall conclusion – no improvement over the standard tagging algorithm (yet)
  - ...but we have more ideas to entertain

# Classifiers and samples

- **Classifiers**

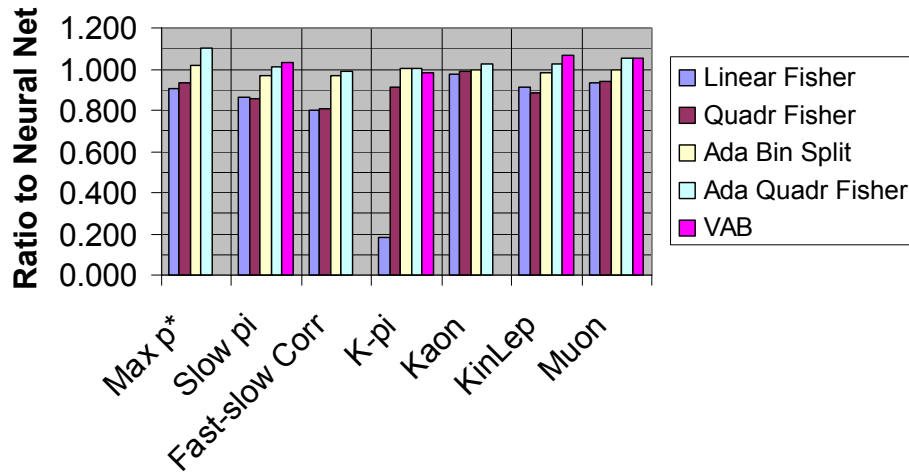
- Linear and quadratic discriminant analysis – L(Q)DA
- Boosted binary splits
- Boosted QDA
- Boosted decision trees
- Random forest with one variable randomly selected for each decision split
- Voronoi Adjusted Boundary
  - see my talk in the parallel statistics session in December 2004
- All (except VAB) described in physics/0507143 and references therein

- **Samples for subtaggers**

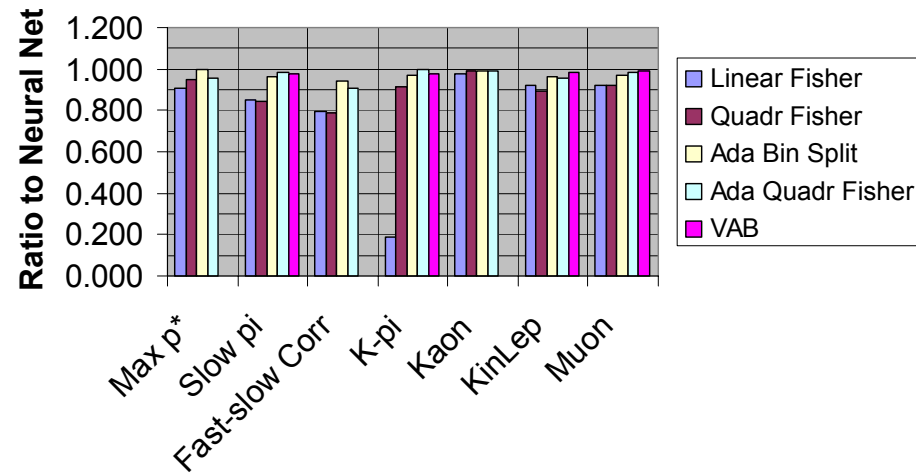
- 4-7 dimensions (a low-dimensional problem by current statistical standards)
- 5k-700k points in the training sample

# Subtaggers (early version of SPR, early 2005)

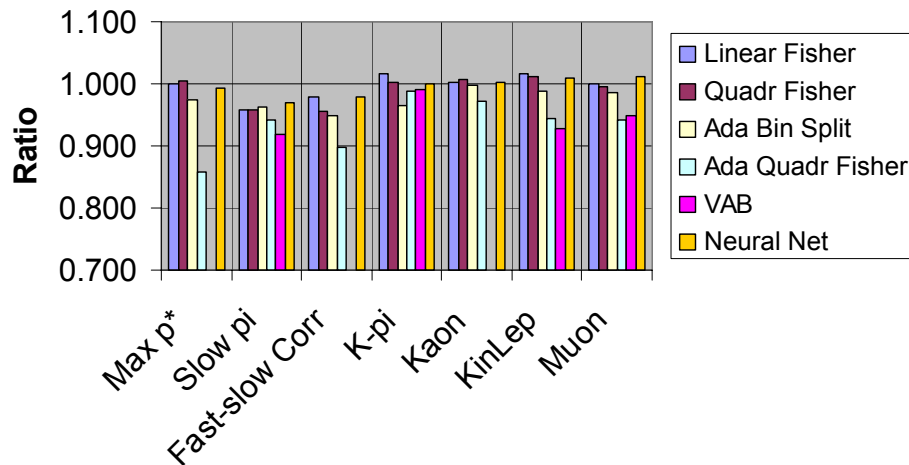
Training Q



Test Q



Test/Training

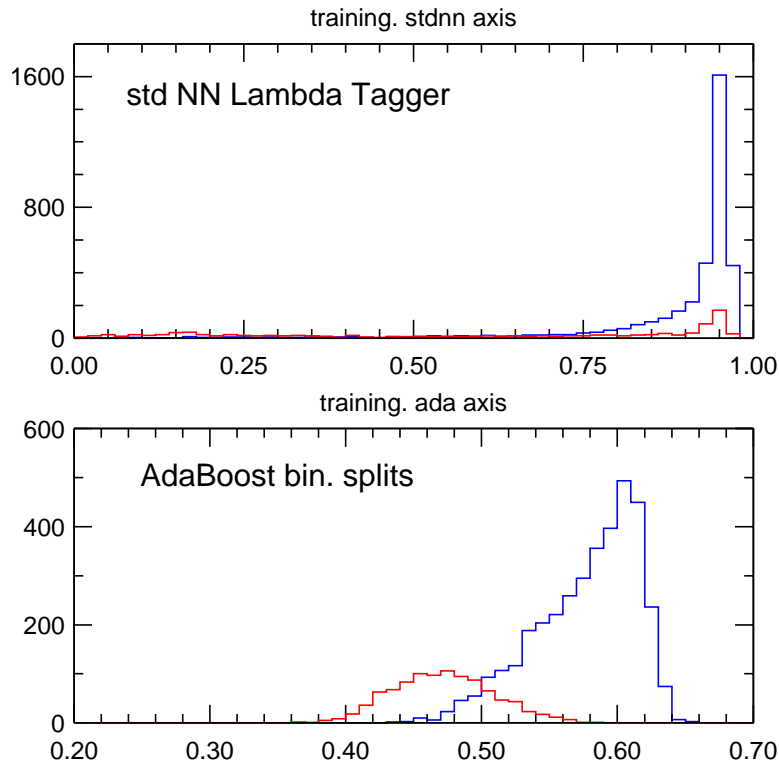


## Summary:

- **Neural Net is surprisingly robust – gives practically the same Q on training and test samples (somewhat in disagreement with my previous NN experience)**
- **For every subtagger, SPR offers a method that performs as well as NN. Among all classifiers, NN offers the best overall performance.**

# Lambda tagger

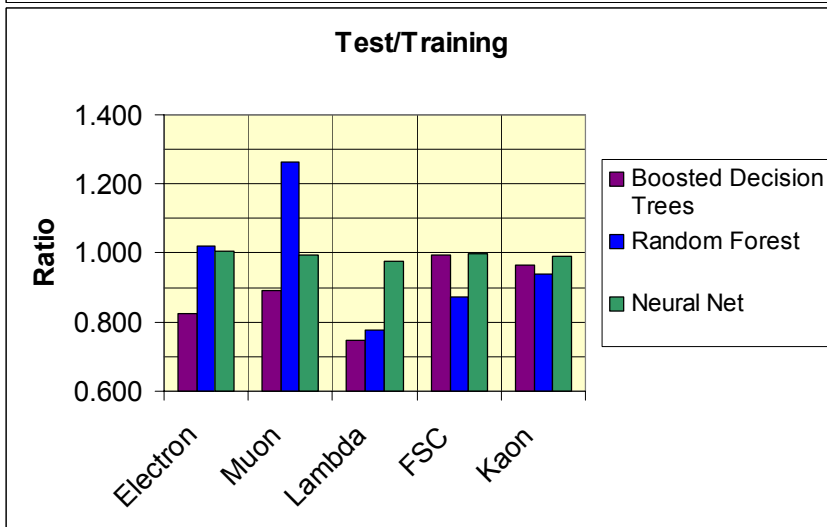
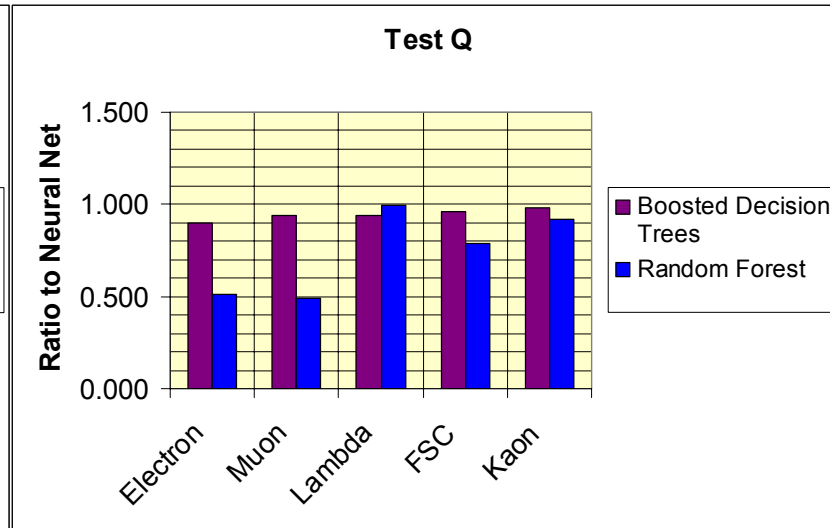
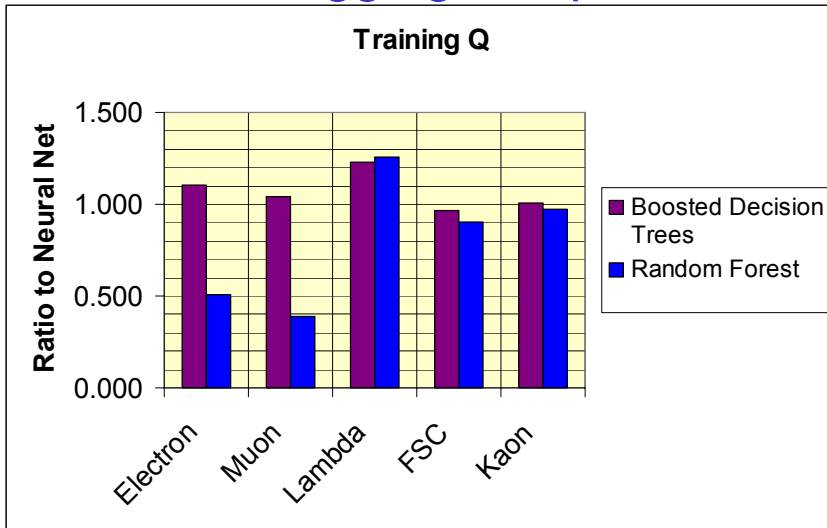
Separation of true Lambdas from background (4600 training points in 6 dimensions)



Something went wrong in the neural net training... Maarten retrained and NN output looks more reasonable now.

# Subtaggers, a quick exercise this week

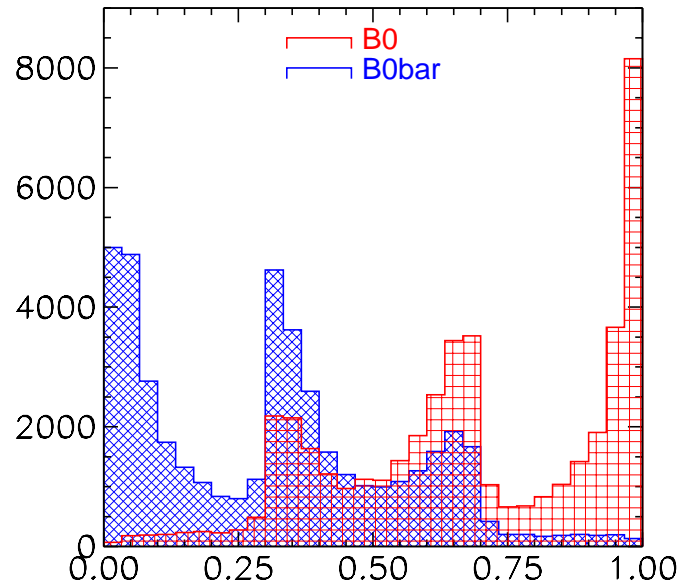
- In early 2005, boosted decision trees and random forest were not yet implemented
- Subtagging samples are different from those in the earlier exercise



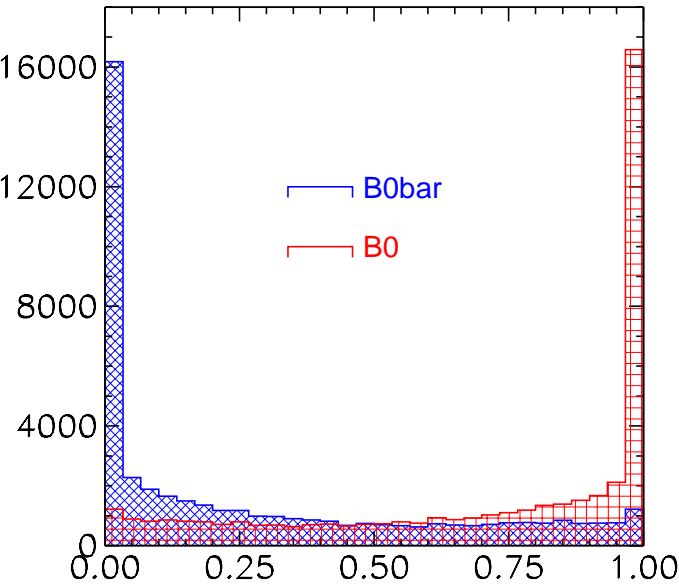
- **NN still gives the best overall performance**
- **Random Forest performs quite poorly for the electron and muon subtaggers, gives a decent but hardly optimal Q for FSC and Kaon, and works well for Lambda. (Where is Breiman now with his “Random Forest is unexcelled in accuracy”?)**
- **Disclaimer: AdaBoost and Random Forest parameters have not been fine-tuned. A quick and dirty exercise!**

# Electron tagger (as an example)

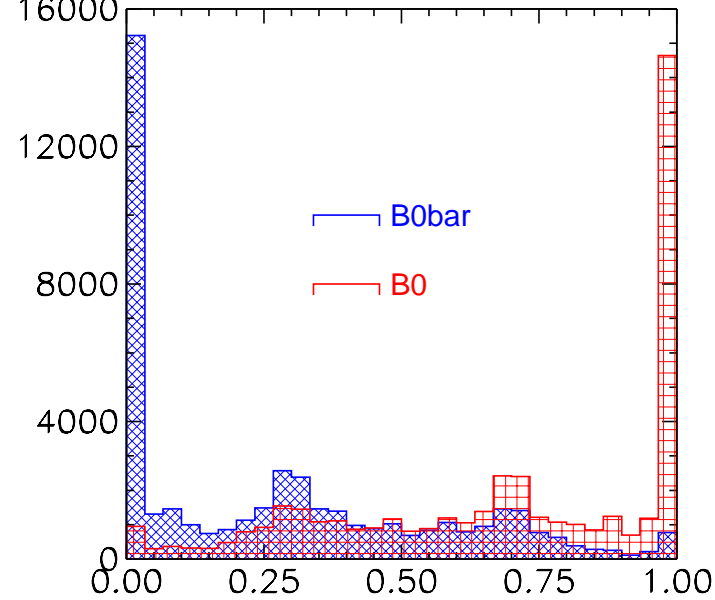
neural net in 4 dimensions



boosted decision trees in 4 dimensions



random forest in 4 dimensions



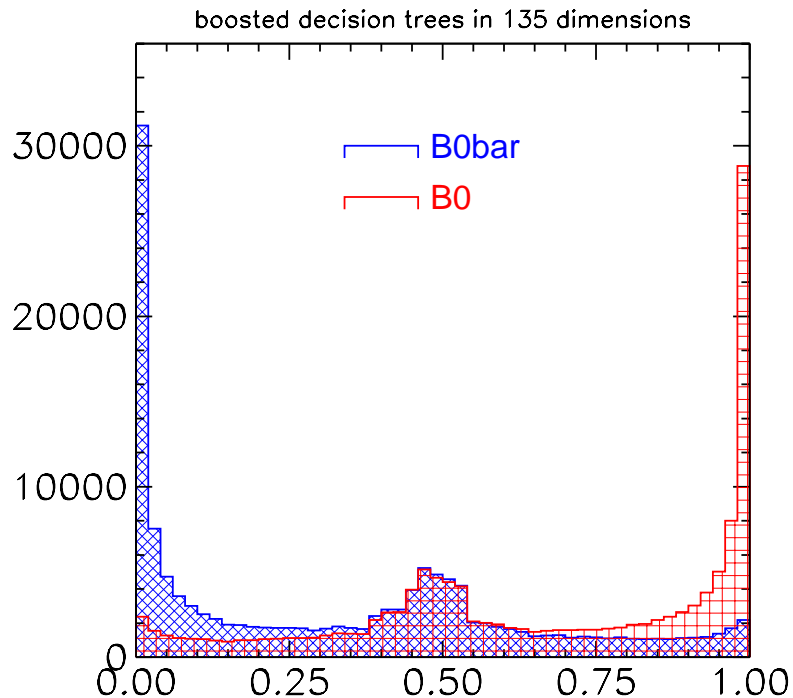
# What can we learn from analysis of subtagger samples?

- Not surprisingly, Neural Net works quite well in low-dimensional problems in the absence of irrelevant or strongly correlated inputs
- Boosted decision trees and Random Forest are classifiers for high-dimensional problems. You can't fully appreciate their power with low-dimensional samples.
- Boosted binary splits are a very quick, robust and powerful classifier!
- Trying to find a new classifier than can squeeze a few more percent out of the same low-dimensional input data is not the way to go. Let us use the power of multivariate classification engines and impose no constraints on the number of input variables!

# Alternative tagging algorithms

- Akram and I started a more serious effort in August
- First exercise:
  - No subtaggers: Dump all remotely interesting variables into one big ascii file and use boosted decision trees to separate  $B^0/B^0_{\text{bar}}$
  - 8 tracks per event selected: electron, muon, 3 kaons and 3 slow pions
  - 11 variables per track:  $p^*$ , charge,  $\cos\theta_{\text{Miss}}$ ,  $\cos\theta_{\text{ThrustAngle}}$ ,  $\text{docaXY}$ ,  $\text{docaZ}$ ,  $\text{dedxSvt}$ ,  $\text{dedxDch}$ ,  $E_{90W}^*$ ,  $n_{\text{SvtHits}}$ ,  $n_{\text{DchHits}}$
  - PID quantities for 5 tracks (electron, muon, stiffest kaon and two softest pions):  $\text{pidKDrc}$ ,  $\text{pidPiDrc}$ ,  $\text{pidPrDrc}$ ,  $\text{pidKDch}$ ,  $\text{pidPiDch}$ ,  $\text{pidPrDch}$
  - fast-slow correlation quantities for 3 slow pions:  $\text{LiKSlow}$ ,  $\text{pStarFast}$ ,  $\cos\theta_{\text{SlowFast}}$ ,  $\cos\theta_{\text{ThrustFast}}$
  - a few global quantities:  $n_{\text{Ks}}$ ,  $n_{\text{Lambda}}$ ,  $n_{\text{Ele}}$ ,  $n_{\text{GammaConv}}$ ,  $\text{sumPt}$
  - as Maarten pointed out, we should have included  $\text{maxpstar}$  track as well

# Results and outlook



**B0/B0bar tagging at BaBar by boosted decision trees with 135 input variables.**

**Time needed to train 50 boosted trees on 500k events = 1+ day.**

**Q=25.2% (10 bins on test data).**

**Not as good as the official 30%, but hey – this is just a first exercise.**

**Two more exercises in the near future:**

- 1) Add maxpstar track and remove quantities that were found useless in the first optimization. Re-run on many dimensions.**
- 2) Add more variables to subtaggers.**

**Suggestions are welcome!**

# Summary

- With the input variables already used for official tagging, finding a way for improvement would be hard
- How much can one gain by adding more variables and using powerful multivariate classifiers? Work in progress. Hopefully more results at future BaBar meetings.