Research on Event Search

Andrey Ustyuzhanin
Yandex, NRC «Kurchatov Institute» Moscow,
Imperial College, London
Quest for analysis sensitivity

Analysis Value Chain

Get datasets (Real, MC, ...)

Pre-selection

Pre-processing (e.g., add variables)

Event selection

Counting/fitting

Systematics Estimation

Significance Estimation

train → test

cut-based

MVA-based
Sources of better sensitivity

1. more powerful algorithms (e.g. BDT, Deep Neural Networks)
2. improved features (e.g. «isolation» variables or particle identification)
3. complex training scenarios (e.g. n-folding, ensembling, blending, cascading)
Price for sensitivity

How do I check quality of event discriminating function?
- Overfitting?
- Correlations?
- Relevance of figure of merit to analysis significance?

How do I deal with complexity?
- Estimate influence of model parameters
- Extra computation
- Organization (cross-checks, collaboration)
MVA Performance (ROC, Learning curve)

Decision Tree

Overfitting

Underfitting

RandomForest

ROC comparison

Learning curves train A test B

Training set accuracy

Test set accuracy

Number of iterations

Training set accuracy

Test set accuracy
MVA algorithms: easy to find, hard to choose

Families:
- Boosted Decision Trees (BDT)
- Artificial Neural Network (ANN)
- Support Vector Machine (SVM)
- Clustering, Bayesian Networks, ...

Implementations
- TMVA (60+ algorithms)
- NeuroBayes
- python scikit-learn
- R packages
- Private (Matrixnet, predict.io)
- XGBoost, ...
Figure-of-Merits Land

- Area under ROC
- Likelihood
- Misclassification
- False Positive, False Negative
- Punzi measure
  \[
  \frac{S}{\sqrt{S+B}}, \frac{S}{\sqrt{B}}, \ldots
  \]
Complexity indicators

‘I can’t remember which version of the code I used to generate figure 13’

‘The new student wants to reuse that model I published three years ago but he can’t reproduce the figures’

‘I thought I used the same parameters but I’m getting different results!?’

‘It worked yesterday!’

‘Why did I do that?’

‘Where are events selected with previous version of reconstruction software?’
Complexity sources

- Domain (Physics)
- Datasources & formats
- Analysis step details (algorithms)
- (Distributed) team communication
Analysis complexity

Case: $\tau \rightarrow 3\mu$ (LHCb)

Get datasets (Real, MC, ...)

Pre-selection

Pre-processing (e.g., add variables)

Event selection

Counting/fitting

Systematics Estimation

Significance Estimation

Repeat count:

$10^2$  $10^2$  $10^3$  $10^2$  $10^2$  $10^2$

Trained models: $\sim 1500$

Requires dedicated framework!
Reproducible Experiment Platform (REP)

software infrastructure to support a collaborative ecosystem for computational science. It is a solution for team of researchers that allows

- running computational experiments on big shared datasets,
- obtaining reproducible and repeatable results,
- comparing measurable result consistently.
REP features/requirements

1. research automation, i.e. defining modules that can be reused later on,
2. consistent automatic cross-check,
3. online visually enhanced shared interactive environment,
4. result reproducibility (code/data provenance),
5. support for existing standard modules,
6. scalability (performance increase as additional [hardware] resources are available),
7. [flat learning curve]
Web Search Workflow

User, query → inverse index → Ranking Model

Top Results:
1. 2. 3.

Iterative model selection/creation

- Feature Design
- Model selection
- Significance Estimation
Collaborative work redux

- Total «freedom»
- Formal agreements
- Experiments repository
  - share of experience, source code reuse
  - data specification, parameters, version
- Regulative infrastructure
- Automated hypotheses testing
  - 10s per week \(\Rightarrow\) 1000s per week

CVS
Programming language
Fixed platform

Continuous Integration
Deployment
Beta-testing
Skills for a physicist

New kind of experimental physicist

- Save time
- Increase team productivity
- Reduce frustration
- Increase chances of employment
REP for HEP

- Online & Interactive
- Support for ROOT & Python & TMVA
- Support for 3rd party classifier (e.g. Matrixnet and SKLearn)
- Run heavy jobs on cluster

Code Example

```python
import train_strategy

folding_scheme = train_strategy.TrainStrategy(directory=work_dir + 'folding/', classifier_type='TMVA')
folding_scheme.set_params(nfolds=10, features=variables, spectators=['mass'])
folding_scheme.fit(train_data_description)
folding_scheme.predict(test_file)

report = folding_scheme.get_model_report()
```

Cases

\[ B_s \rightarrow \mu^+ \mu^- \]
\[ B_s \rightarrow 4\mu \]
\[ \tau \rightarrow 3\mu \]
\[ B \rightarrow K^* \mu^+ \mu^- \]
\[ \ldots \]
Instead of Conclusion

- New source of tools & metrics: **data science**
  - ...as well as source of complexity
- Research reproducibility = defeat of complexity
  - Status: looking for new cases, adopters
- How to try?
  - Hands-on introduction **tomorrow at 16:15**
  - andrey.ustyuzhanin@cern.ch
Backup
N-folding, training scheme example
(works well for limited statistics)

Split data in N folds randomly

Take i-th fold, train formula on remaining folds, apply to selected one

See the difference