Pitfalls of evaluating a classifier's performance in high energy physics applications

Gilles Louppe, NYU (@glouppe)
Tim Head, EPFL (@betatim)

December 11, 2015
ALEPH workshop, NIPS

Disclaimer

The following applies only for the learning protocol of the *Flavours of Physics* Kaggle challenge.

See notebook for further details.

Flavours of Physics: Finding $\tau \mapsto \mu\mu\mu$ challenge

Given a learning set $\mathcal{L}$ of

- simulated signal events $(\mathbf{x}, s)$
- real data background events $(\mathbf{x}, b)$,

build a classifier $\varphi : \mathcal{X} \mapsto \{s, b\}$ for distinguishing $\tau \mapsto \mu\mu\mu$ signal events from background events.

Control channel test

The simulation is not perfect: simulated and real data events can often be distinguished.

To avoid exploiting simulation versus real data artefacts to classify signal from background events, we evaluate whether φ behaves differently on simulated signal and real data signal from a control channel $\mathcal{C}$.

Control channel test: Requires the Kolmogorov-Smirnov test statistic between $\{\varphi(\mathbf{x})|\mathbf{x} \in \mathcal{C}^{\text{sim}}\}$ and $\{\varphi(\mathbf{x})|\mathbf{x} \in \mathcal{C}^{\text{data}}\}$ to be strictly smaller than some pre-defined threshold t.

Proposition

Assuming that

- control data can be distinguished from training data,
- simulated features are more discriminative than they are in real data,

Then, even by chance, φ might exploit simulation versus real data artefacts to classify signal from background events, while still passing the control channel test.

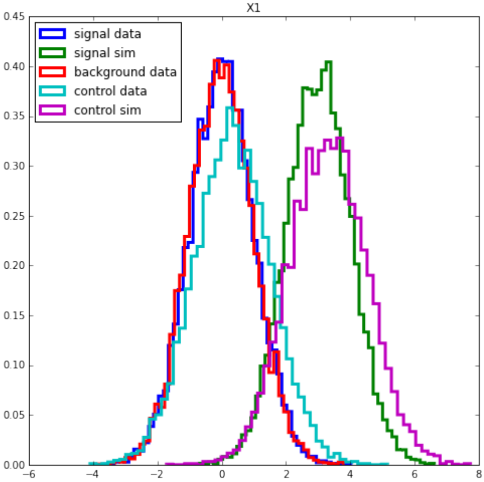
Therefore,

- The true performance of φ on real data may be significantly different (typically lower) than estimated on simulated signal events versus real data background events.
- Passing the KS test does not tell you anything about φ.

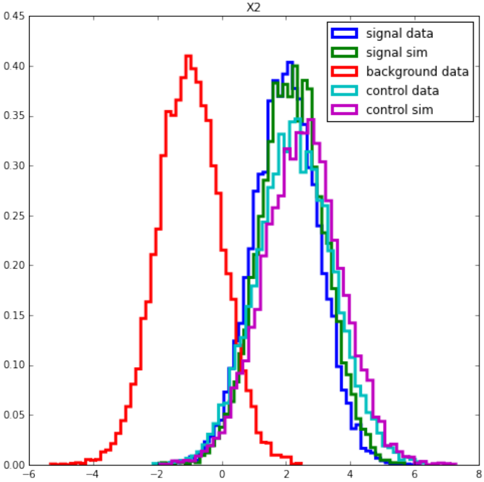
Toy example

Let us consider an artificial classification problem between signal and background events, along with some close control channel data $\mathcal{C}^{\text{sim}}$ and $\mathcal{C}^{\text{data}}$.

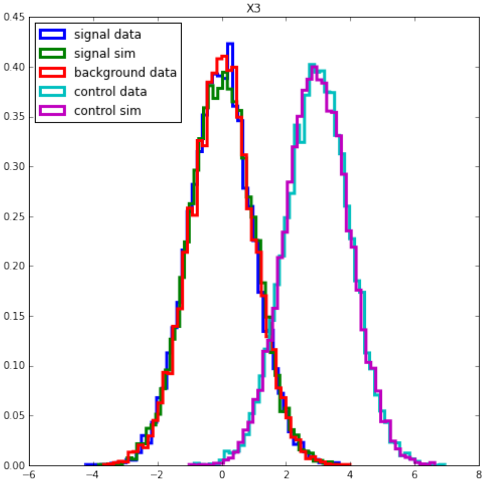
Let us assume an input space defined on three input variables X_1, X_2, X_3 as follows.



X_1 is irrelevant for real data signal versus real data background, but relevant for simulated versus real data events.



X_2 is relevant for background and non-background events.



X_3 is relevant for training versus control events, but has otherwise no discriminative power between signal and background events.

Random exploration

```
from sklearn.ensemble import ExtraTreesClassifier

def find_best_tree(X_train, y_train, X_test, y_test,
                   X_data, y_data, X_control_sim, X_control_data):
    best_auc_test, best_auc_data = 0, 0
    best_ks = 0
    best_tree = None

    for seed in range(2000):
        clf = ExtraTreesClassifier(n_estimators=1, max_features=1,
                                   max_leaf_nodes=5, random_state=seed)
        clf.fit(X_train, y_train)
        auc_test = roc_auc_score(y_test, clf.predict_proba(X_test)[:, 1])
        auc_data = roc_auc_score(y_data, clf.predict_proba(X_data)[:, 1])
        ks = ks_statistic(clf.predict_proba(X_control_sim)[:, 1],
                          clf.predict_proba(X_control_data)[:, 1])

        if auc_test > best_auc_test and ks < 0.09:
            best_auc_test = auc_test
            best_auc_data = auc_data
            best_ks = ks
            best_tree = clf

    return best_auc_test, best_auc_data, best_ks, best_tree
```

Random exploration

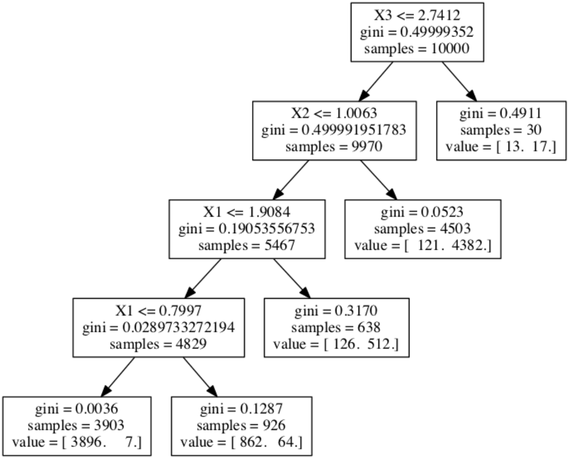
```
auc_test, auc_data, ks, tree = find_best_tree(X_train, y_train,
                                              X_test, y_test,
                                              X_data, y_data,
                                              X_control_sim, X_control_data)

# Estimated AUC (simulated signal vs. data background)
>>> 0.986357983199

# True AUC (data signal vs. data background)
>>> 0.90973817

# KS statistic
>>> 0.0578  # < 0.09
```

What just happened? By chance, we have found a classifier that

- has seemingly good test performance;
- passes the control channel test that we have defined.

This classifier appears to be exactly the one we were seeking.

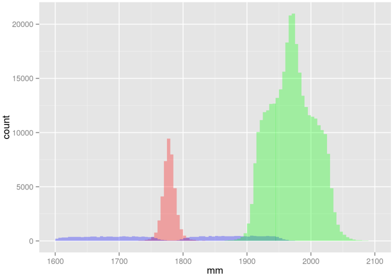
Wrong. The expected ROC AUC of 0.91 on real data signal and real data background is significantly lower than our first estimate, suggesting that there is still something wrong.



φ exploits X_1, i.e. simulation versus real data artefacts to indirectly classify signal from background events, while still passing the control channel test because of its use of X_3!

Winning the challenge

1. learn to distinguish between training and control data,
2. build a classifier on training data, with all the freedom to exploit simulation artefacts,
3. assign random predictions to samples predicted as control data, otherwise predict using the classifier found in the previous step.



A better protocol

If differences between simulated and real data events are fixed, then the problem goes away.

One way to do it is to learn a transformation (e.g., a reweighting) from simulation onto real data from the control channel, and then learn on transformed simulated signal events versus real data background events.