Machine Learning in HPS

HPS Spring Collaboration Meeting 2018

Introduction

- Machine learning is quickly proving a powerful tool in physics, and may be useful to HPS specifically.

- Machine learning algorithms can detect complex patterns and produce rule sets for selecting data that are much more complicated than humans can generally manage.

- In HPS, this could be used for tasks such as optimizing selection of trident particles or improving cluster/track matching.

Weka Workbench

- The Weka Workbench is a Java-based piece of software that was developed by the machine learning group at the University of Waikato in New Zealand.
- It implements a wide range of machine learning algorithms and techniques, with the option to include more through plug-ins.
  - All of the algorithms can be easily run over data from the workbench, without having to modify it or implement separate code bases to try a different methodology.
  - This makes it easy to compare algorithm performance.

Weka Workbench

- Weka is also Java-based, which means that it is easier to integrate its algorithms with HPS-Java, if this is needed.
- Weka uses a simple text-based data format (.ARFF) as input.
- ARFF files are simple.
  - The format of data in ARFF file is defined with one or more "@attribute" declarations.
  - Attributes can be either numbers or user-defined text options.

```
@relation cluster_track_matching
@attribute cluster_rx numeric
@attribute cluster_ry numeric
@attribute cluster_ix numeric
@attribute cluster_iy numeric
@attribute cluster_energy numeric
@attribute cluster_hit_count numeric
@attribute track_rx numeric
@attribute track_ry numeric
@attribute track_px numeric
@attribute track_py numeric
@attribute track_momentum numeric
@attribute track_chi2 numeric
@attribute track_charge { positive, negative }
@attribute track_top_bottom { top, bottom }
@attribute dx numeric
@attribute dy numeric
@attribute dr numeric
@attribute dE numeric
@attribute matched { true, false }
```

Weka Workbench

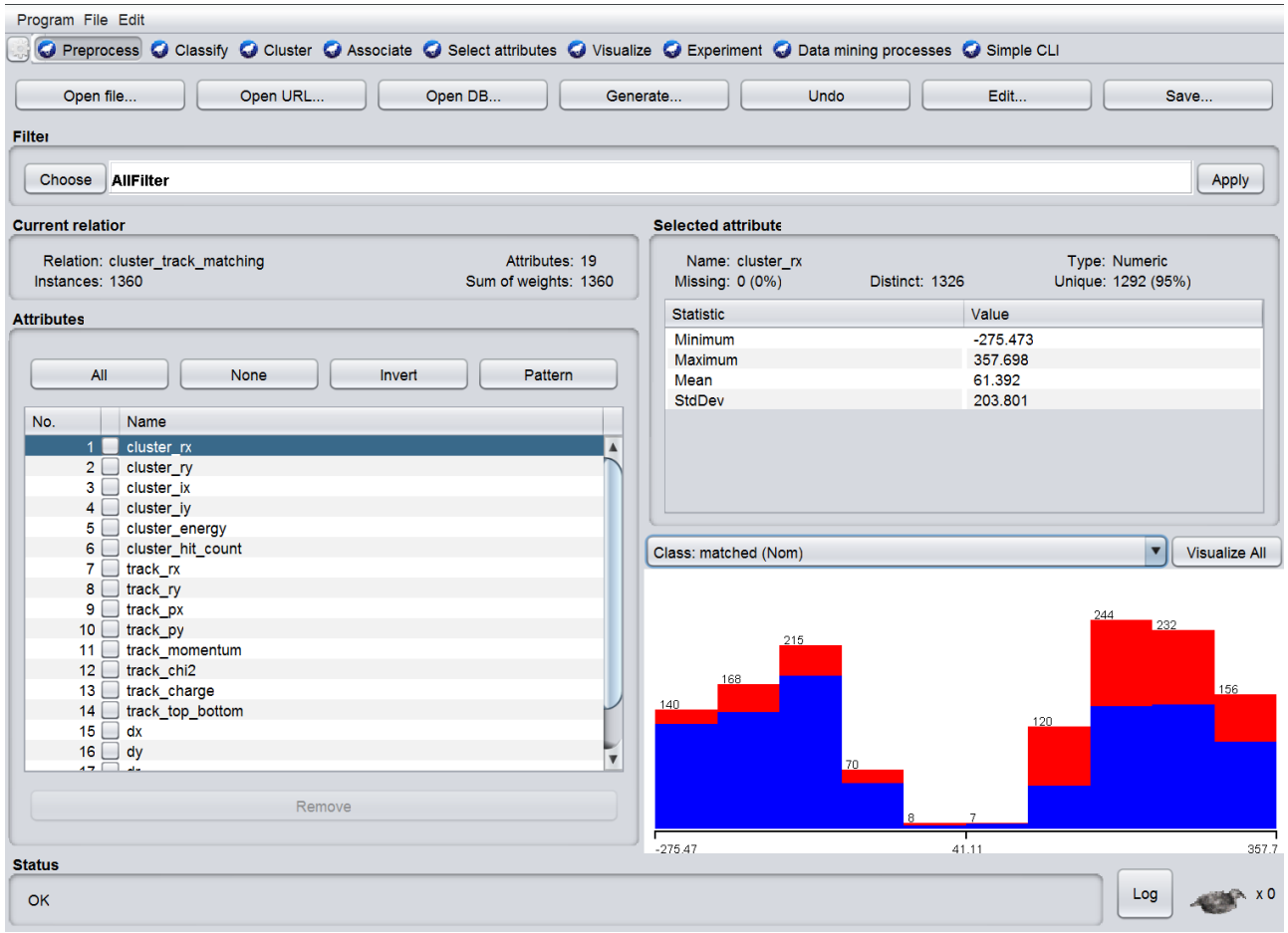
- Data is then entered as “instances.”
  - A data instance is a comma-delimited string, terminated by a newline, which lists all of the attributes, in order.
  - Attributes can be listed as unknown if the value is not available (though not all algorithms can support this).

```
279.992206,-70.582537,18,-4,0.584745,2,274.966112,-21.703473,0.563843,0.007047,0.563952,14.386774,positive,bottom,5.026094,48.879064,49.156112,0.020793,true
-80.957701,43.328051,-9,2,0.822547,4,-79.344136,39.664180,0.790226,0.055653,0.792499,3.935633,negative,top,1.613566,3.663871,4.007253,0.030049,true
265.781747,-84.298850,17,-5,0.531858,2,269.357513,-84.022091,0.559798,0.004270,0.560782,5.787946,positive,bottom,3.575766,0.276759,3.842601,0.028924,true
356.058226,-55.129779,23,-2,0.470472,4,346.273833,-55.754284,0.487205,0.014949,0.487794,1.581284,positive,bottom,9.784393,0.624504,9.948421,0.017323,true
-236.292719,52.460277,-20,3,0.529349,4,-232.830090,57.347875,0.537074,0.013130,0.537666,5.651695,negative,top,3.462628,4.887598,6.170397,0.008317,true
-239.442292,82.947026,-20,5,0.601016,3,-238.229736,89.184332,0.550138,0.009088,0.551293,5.834414,negative,top,1.212556,6.237306,6.576826,0.049723,true
244.090486,-66.340879,15,-4,0.587508,3,243.600525,-63.287139,0.602869,0.001987,0.603468,17.316894,positive,bottom,0.489962,3.053740,3.267923,0.015959,true
-201.840144,60.532634,-18,3,0.576691,3,-189.158986,59.459818,0.601224,0.020317,0.602096,5.246595,negative,top,12.681157,1.072815,12.789553,0.025405,true
145.601850,-34.056682,8,-1,1.267909,4,145.710981,-33.027086,1.286651,0.032730,1.287424,6.971264,positive,bottom,0.109131,1.029597,1.036496,0.019514,true
307.391810,-84.382675,19,-5,0.751099,2,312.131059,-77.903416,0.735064,0.059014,0.738509,7.165451,positive,bottom,4.739249,6.479259,8.183850,0.012590,true
-88.378065,42.978182,-10,2,1.308728,4,-88.264647,38.038501,1.541607,0.003303,1.542179,2.567126,negative,top,0.113418,4.939681,4.949229,0.233451,true
```

- Outputting these files from HPS-Java is quite straight-forward.

Weka Workbench

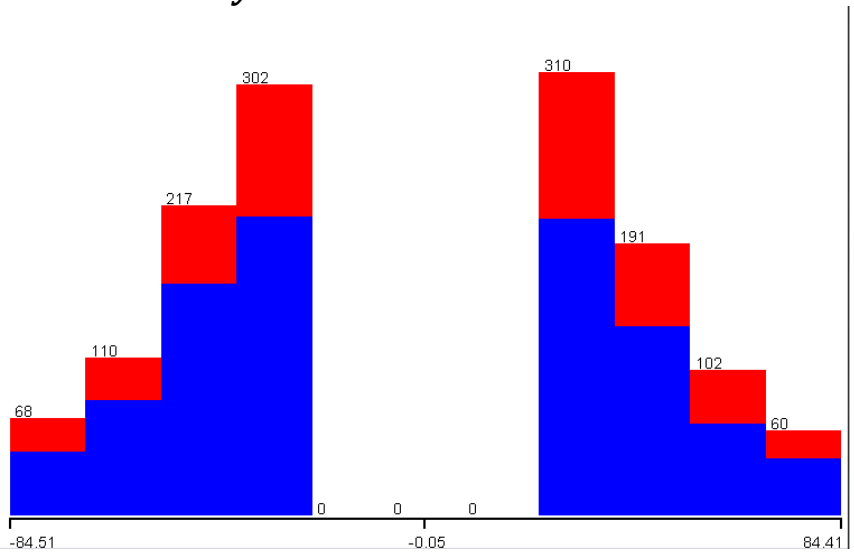
- When an ARFF file is loaded into Weka, it immediately provides a number of visualization tools.



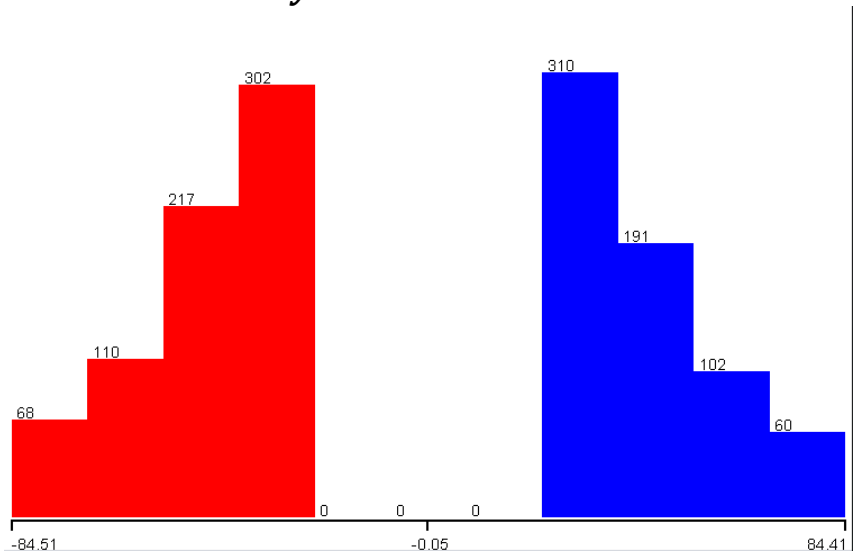
Weka Workbench

- Any numeric attribute may be plotted in terms of any text attribute.

Cluster r_y (Matched/Not Matched)



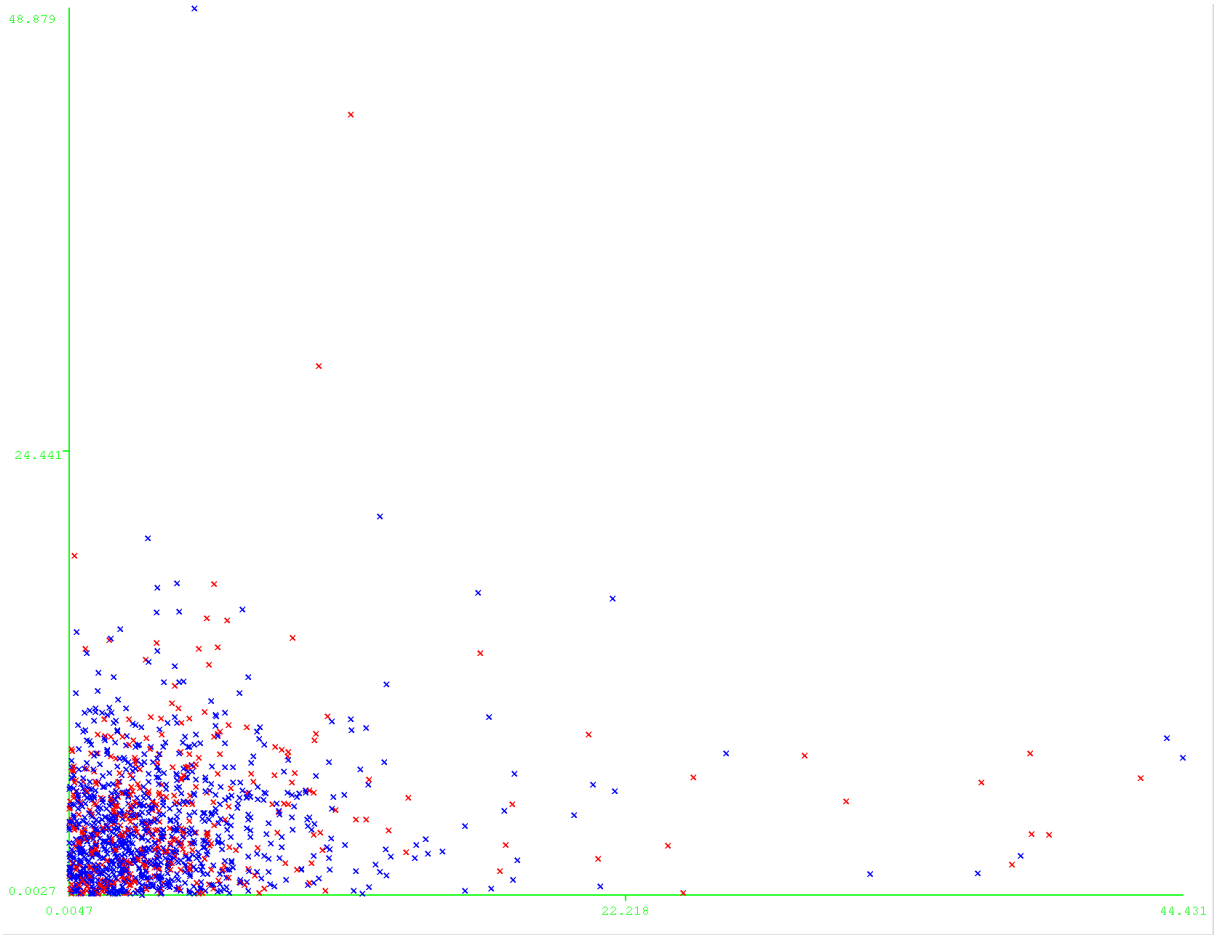
Cluster r_y (Top/Bottom Track)



- This can help visualize which attributes are useful and discriminatory quickly.

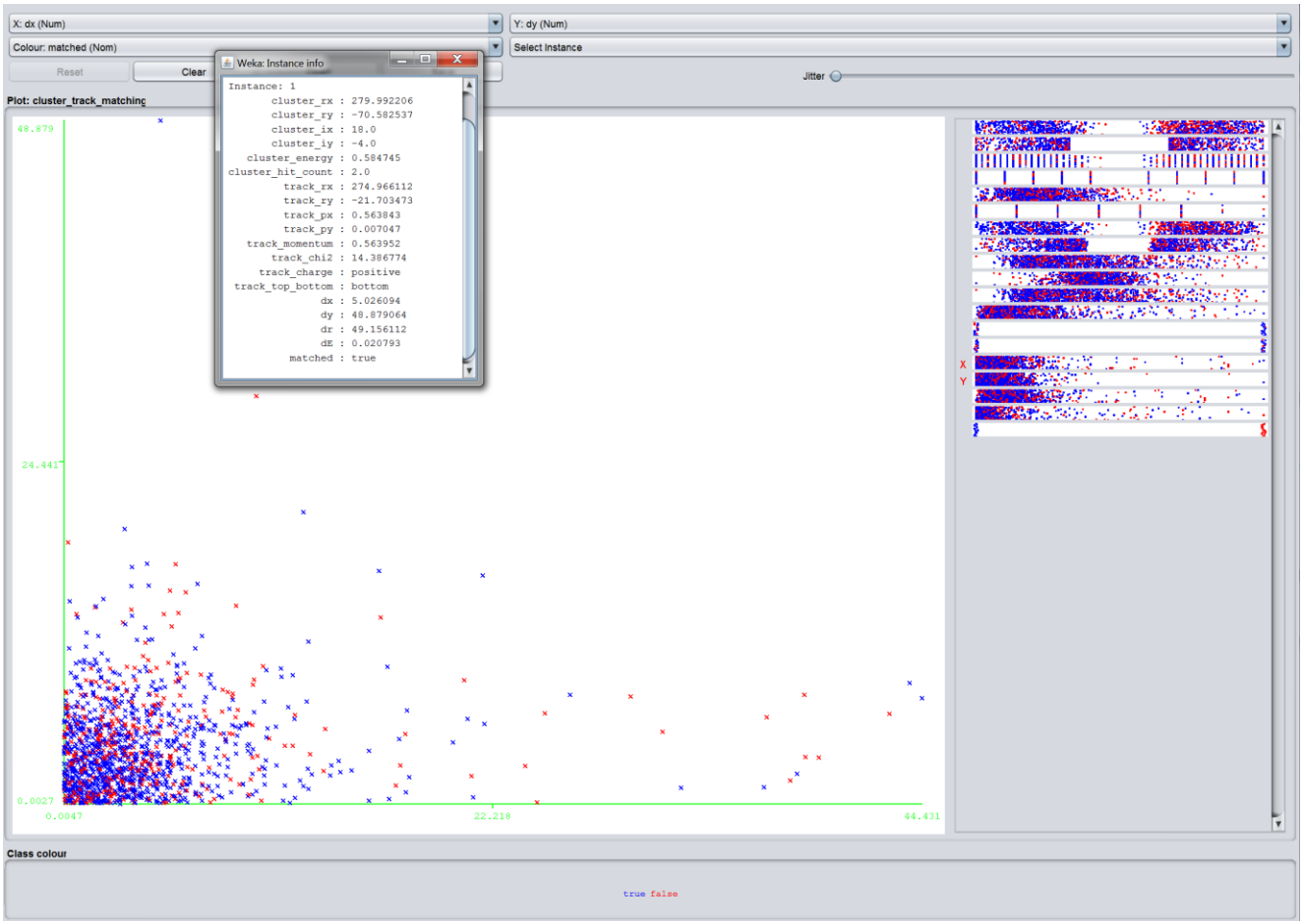
Weka Workbench

- It can also plot any two numerical values on a 2D plot, again against a text attribute.



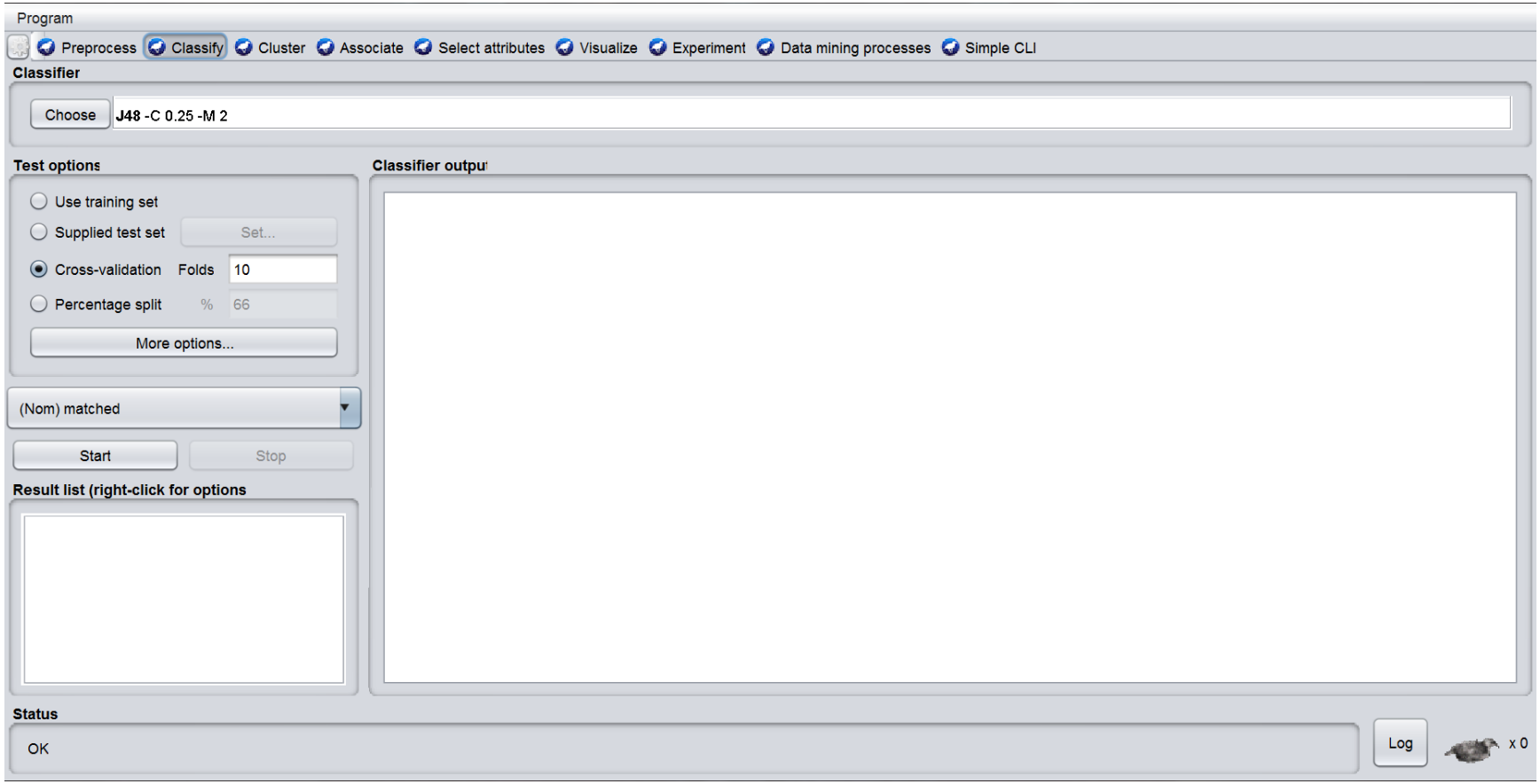
Weka Workbench

- Additionally, on the 2D plot, it is possible to right-click any data point to view its full attribute list.



Weka Workbench

- Weka's "Classify" menu allows machine learning algorithms to be quickly applied to the data set.



Creating a Training Set

- The first, and in many ways, most important task to any machine learning endeavor is generating a training set.

- For HPS, we have the advantage of Monte Carlo.
  - This allows us to generate (hypothetically) as much training data as we need.
  - It also allows us to use generator truth information to make sure that the algorithm always knows the actual, real classification for whatever we are looking at.

- But there is a problem!

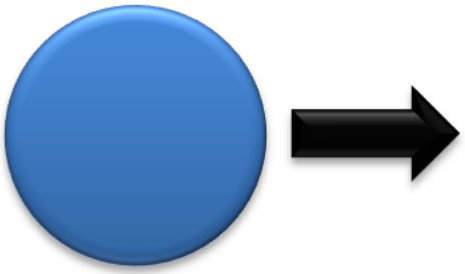
Creating a Training Set

- The HPS-Java software package loses truth information when performing readout.
- In order to access and use this information, we need to make modifications to the readout and reconstruction systems to propagate it forward.
- The SVT truth propagation upgrade was handled by Matt Solt, and will not be discussed here.
- The calorimeter readout has recently been upgraded to also propagate truth information, and will be discussed briefly.

Propagating Calorimeter Truth

- When readout is run in HPS-Java, it takes in SLIC truth hits. These are fairly simple and have a few properties:
  - Channel ID
  - Energy deposition
  - Hit time
  - Energy deposition by truth particle

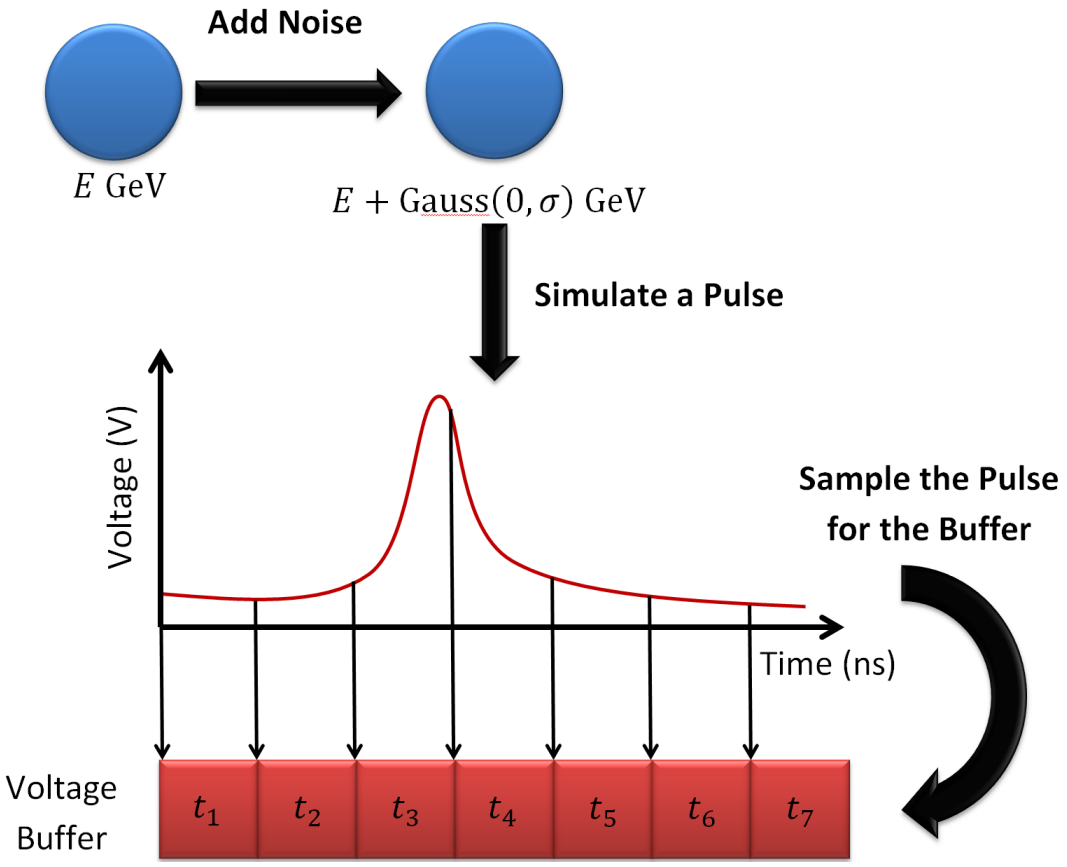
SLiC Truth Hit



- Energy Deposition (E_{truth})
- Time (t)
- Channel ID

Propagating Calorimeter Truth

- Each event, SLIC truth hits are used to generate a pulse, which is dumped into a voltage buffer.



Propagating Calorimeter Truth

- The voltage buffer values are subsequently used to populate an ADC buffer, which can be integrated in the same manner as the hardware in a sample crosses the integration threshold.

- When a trigger occurs, the readout outputs the ADC buffer in a range around the trigger time.

- However, because truth information is lost at the point of the voltage buffer, this means there is no easy way to correlate the SLIC truth hits to the ADC values, or to the hits created from them.

Propagating Calorimeter Truth

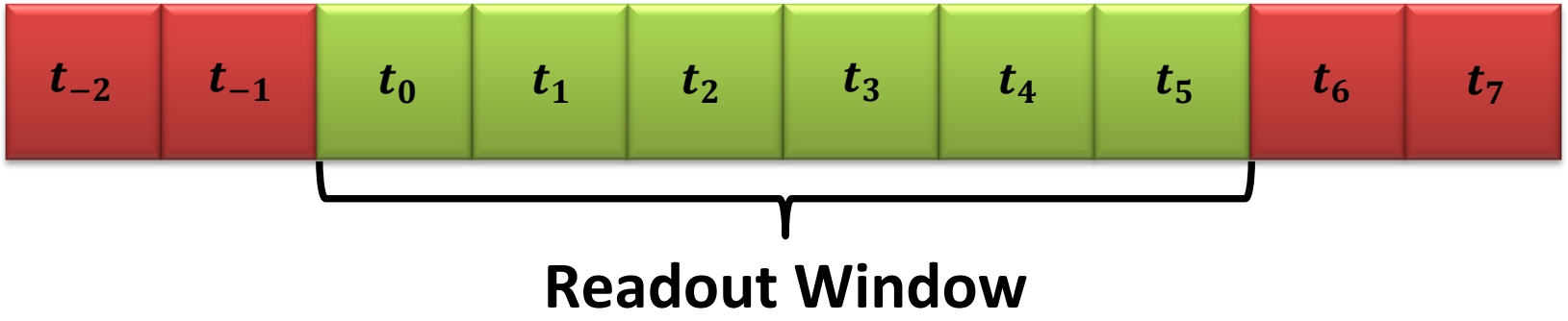
- To account for this, a new buffer is created which stores SLIC truth hits in buffer cells corresponding in time to the ADC buffer.

- When a trigger occurs, the truth hits in the truth buffer are output as well, and LCRelation objects are created to link them to the ADC buffer samples.
  - Note that a slightly larger output range is used for truth hits, to account for the rising time in the pulse emulation.
  - Truth hits are also given timestamps that are relative to the readout window, rather than the beam bunch.

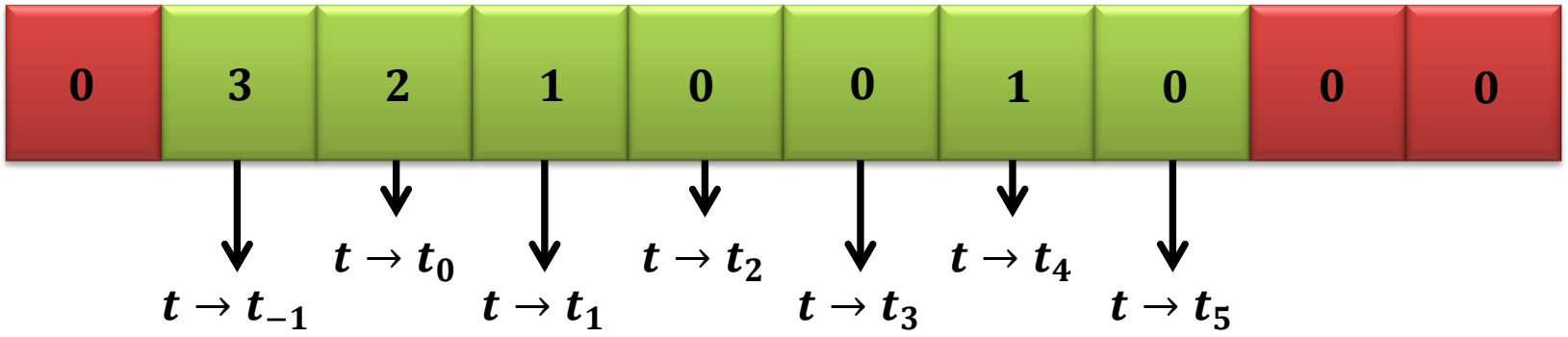
- Then, during reconstruction, these hit relations can be used to determine which hits are connected to the integrated hit generated from the ADC buffer.

Propagating Calorimeter Truth

ADC Buffer:



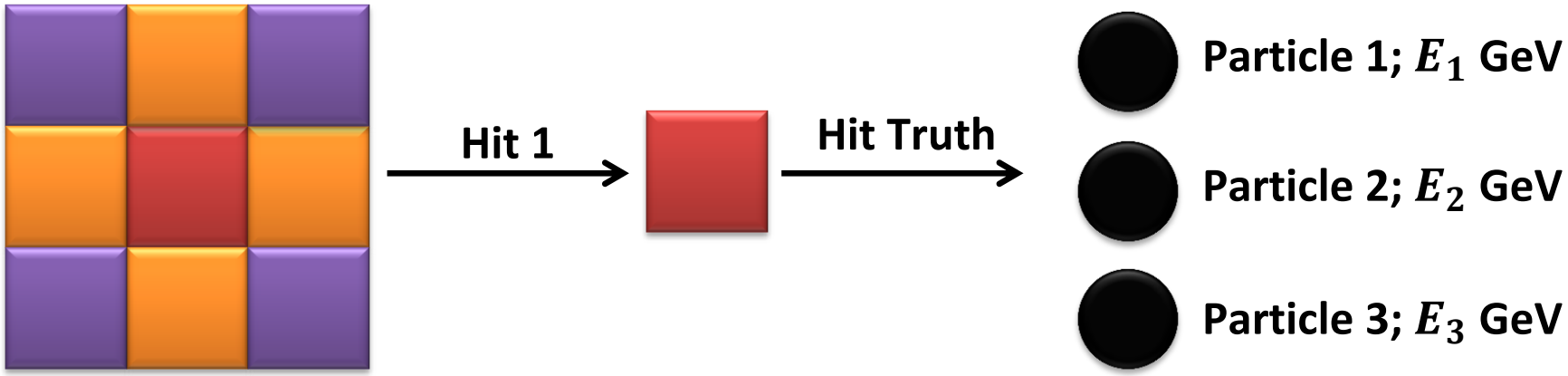
Truth Buffer:



Using Calorimeter Truth Data

- Now that truth information is properly available, we need to develop a method to use it to associate clusters with truth particles.

- Each cluster stores its calorimeter hits.
  - With the new truth propagation, each calorimeter hit is now a "SimCalorimeterHit" object which contains information on each truth particle that contributed to the cluster, and its truth energy deposition.



Using Calorimeter Truth Data

- By using the truth information in each hit, we determine several things for each cluster:
  - The total truth energy deposition.
  - The total energy deposition from each contributing particle.
  - The percentage energy deposition from each contributing particle.

- Using this, we treat any particle that contributes more than a certain percentage of the cluster's energy as a "truth particle" for cluster.
  - We do not select only one particle, as sometimes more than one particle can contribute significantly, making it ambiguous which is the "real" particle.

Creating a Training Set

- Now that truth information is properly available, a training data set can be created.

- As a simpler example, we will look at cluster/track matching.

- Data is selected using the following steps:
  - Pure trident reconstructed data is used.
  - Events are filtered to only consider analyzable events.
    - Two tracks, one positive/one negative and one top/one bottom.
  - Matt Solt's truth algorithm is used to obtain the Monte Carlo truth particle that is associated with each track.
  - Tracks are considered to be "matched" with any cluster that its truth particle contributed more than 25% of the total energy to.

Creating a Training Set

- However, a new problem arises.
- It turns out that the calorimeter truth information links truth particles to very disparate hits across the calorimeter face.
- Additionally, it seems that all of the truth particles in the event contribute to the same set of calorimeter crystals.
- This issue is evident even at the SLIC output level, so it does not appear to be an error in truth propagation.

Creating a Training Set

- A sample SLIC event is shown below.

```
 ************************************************************************
 *** Particle Analysis ************************************************
 ************************************************************************
Particle of type  e+ produced at time t =   0.0 ns and vertex <  -0.0,    0.0,    0.5> with + charge and momentum 0.635 GeV.
      0.001 GeV ( 58.2%) :: Hit at (-16,   1) with energy 0.002 GeV (truth energy 0.002 GeV) at time 5.2 ns.
      0.001 GeV (100.0%) :: Hit at (-15,  -1) with energy 0.001 GeV (truth energy 0.001 GeV) at time 5.2 ns.
      0.000 GeV (100.0%) :: Hit at (-19,  -3) with energy 0.000 GeV (truth energy 0.000 GeV) at time 5.8 ns.
      0.000 GeV (100.0%) :: Hit at ( 20,  -1) with energy 0.000 GeV (truth energy 0.000 GeV) at time 5.5 ns.
      0.000 GeV ( 77.3%) :: Hit at (-15,   1) with energy 0.000 GeV (truth energy 0.000 GeV) at time 4.9 ns.
      0.000 GeV (100.0%) :: Hit at (-16,   4) with energy 0.000 GeV (truth energy 0.000 GeV) at time 5.5 ns.
      0.001 GeV (100.0%) :: Hit at (-21,   1) with energy 0.001 GeV (truth energy 0.001 GeV) at time 5.6 ns.
      0.000 GeV (100.0%) :: Hit at (-17,   3) with energy 0.000 GeV (truth energy 0.000 GeV) at time 5.4 ns.
      0.002 GeV ( 18.1%) :: Hit at (-16,  -1) with energy 0.011 GeV (truth energy 0.011 GeV) at time 5.1 ns.
      0.001 GeV (100.0%) :: Hit at (-17,  -1) with energy 0.001 GeV (truth energy 0.001 GeV) at time 5.3 ns.
Particle of type  e- produced at time t =   0.0 ns and vertex <  -0.0,    0.0,    0.5> with - charge and momentum 0.670 GeV.
      0.001 GeV ( 33.5%) :: Hit at (-16,   1) with energy 0.002 GeV (truth energy 0.002 GeV) at time 5.2 ns.
      0.000 GeV ( 22.7%) :: Hit at (-15,   1) with energy 0.000 GeV (truth energy 0.000 GeV) at time 4.9 ns.
      0.009 GeV ( 81.9%) :: Hit at (-16,  -1) with energy 0.011 GeV (truth energy 0.011 GeV) at time 5.1 ns.
Particle of type  e- produced at time t =   0.0 ns and vertex <  -0.0,    0.0,    0.5> with - charge and momentum 0.995 GeV.
      0.000 GeV (  8.4%) :: Hit at (-16,   1) with energy 0.002 GeV (truth energy 0.002 GeV) at time 5.2 ns.
```

Creating a Training Set

- This issue often causes tracks to be associated with clusters that are obviously wrong.
- It also can cause a cluster to be linked with a trident even though it realistically is not.
- This is an issue, as it pollutes the training data and makes it impossible for the machine learning algorithm to produce a useful rule set.
- Presently, the cause of this issue is not known, but it is under investigation.

Summary

- Machine learning may be useful for optimization HPS analysis algorithms.

- Truth information in MC is a useful tool for producing training data sets.
  - Truth information was previously not available at the reconstruction stage.
  - This was rectified for the calorimeter by me.
  - This was corrected in the SVT by Matt Solt.

- SLIC appears to be doing something strange when reporting truth energy depositions for calorimeter truth hits.
  - This makes it impossible to properly link calorimeter objects with SVT objects due to creating extraneous, false relations.
  - The cause is unknown and is under investigation.