Energy Calibration with ML

Jason Bryslawskyj

October 22, 2018

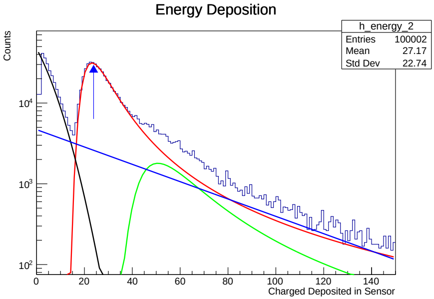
Energy Calibration

- Sensors in EmCal calibrated by looking for minimum ionizing particles (MIPs)
- Charge deposited by MIPs follows Landau distributions:

$$\text{Signal} = \text{Landau}_0(\mu_0, \sigma_0) + \text{Landau}_1(\mu_0, \sigma_0)$$

- Shown as Red, Green
- Background is composed of a Gaussian pedestal (Black)
  + high Energy particles (exp,blue), which are signal for analysis, but BG for calibration
- Regression complicated by zero-supression, which cuts a square notch in a random location between 0 and 10
- Challenge: Fit 200,000 sensors, all with differently shaped signal and background
- Find Most Probable Value (MPV) of MIP Landau (Blue Arrow)

Example of charge disposited in single sensor



Monte Carlo Training and Testing Data Generation

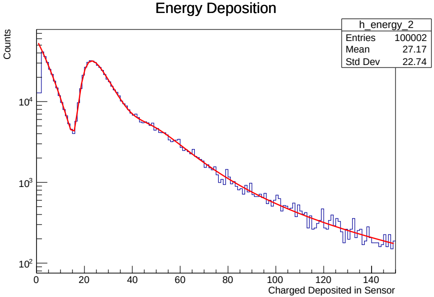
- Training and Testing data generated with Monte Carlo in
  generate_data.C
- Signal + Background Function (Red):

$$\text{Signal} = \text{Landau}_0(\mu_0, \sigma_0) + \text{Landau}_1(\mu_0, \sigma_0)$$

$$+\text{Gauss}(\mu_1, \sigma_1) + \exp(-\tau)$$

- Training and testing histograms (blue) generated by randomly sampling MC function 100000 times

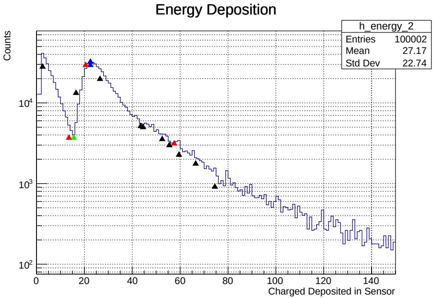
Example of charge disposited in single sensor



Feature Selection

- Naively fitting using regression with entire underlying functional form fails
- Due to the large number of fit parameters
- Can fit after seeding using features discribing histograms shape
- Find following features:
- Local Minima (Green) and Maxima (Blue)
- Locations where $\frac{dy}{dx} = 0$ (Red)
- Locations where $\frac{d^2y}{dx^2} = 0$ (Black)
- Performed in get_features.C

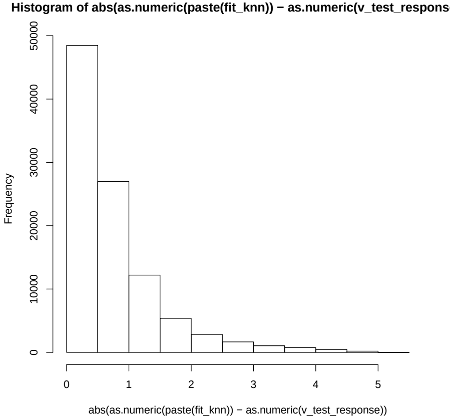
Example of charge disposited in single sensor



Finding MIP using kNN

- Find MIP location (MPV) using k-Nearest Neighbors
- Feartures used as predictors
- Predictors: Minima, Mixima, $\frac{dy}{dx} = 0$
- MIP location (MPV) used as response
- Success measured by distance between true MIP location and predicted MPV
- Fraction of predictions closer that 1 unit from true MIP:
  - For K=1: 75.5%
  - For K=2: 74.4%
  - For K=15: 68.8%
  - For K=100: 60.9%
- Performed in find_mpv_KNN.R

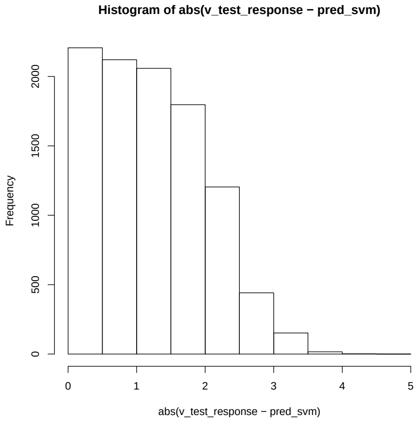
K =1 Difference between true and predicted MIP location (Lower is Better)



Finding MIP using SVM

- Find MIP location (MPV) using a Support Vector Machine
- Feartures used as predictors
- Predictors: Minima, Mixima, $\frac{dy}{dx} = 0$ and $\frac{d^2y}{dx^2} = 0$
- MIP location (MPV) used as response
- Success measured by distance between true MIP location and predicted MPV
- Fraction of predictions closer that 1 unit from true MIP:
  - For Radial Kernal with Cost=1: 43.3%
  - For Radial Kernal with Cost=0.1: 42.3%
  - For PolyN Kernal with Cost=1: 42.3%
  - For PolyN Kernal with Cost=1: 40.1%

K =1 Difference between true and predicted MIP location (Lower is Better)



Finding MIP using a Neural Network

- Finding MIP using a Neural Network
- Uses bin contents of histograms directly instead of feature space
- Input layer: bin contents of histograms
- 2 hidden layers with 10 neurons, 8 neurons
- Ouput layer: MPV location
- Fraction of predictions closer that 1 unit from true MIP: 69.7%

K =1 Difference between true and predicted MIP location (Lower is Better)