Module 4
Particle Physics - Data analysis Lab

M. Toharia (J.Trudeau)

Introduction

In order to search for the Higgs particle, physicists collide protons against one another. Very rarely, a Higgs particle is produced from a collision event. Once the Higgs is produced, it immediately decays. For every 1000 Higgses produced, only two decay into photons. But photons are easy to distinguish compared to other decay products (hadronic jets).

After a run of collisions, physicists extract from the full record of collision events the events that contain two high energy photons (these could come from a Higgs decay or from a much less exciting process).

Our job is to analize 500 000 diphoton events recorded and search a Higgs particle within them. These events are not true data, but have been simulated with particle physics event generators and look very much like the real events at the LHC.

The data file contains some $2 \sim 3$ million lines of data.

Task (A)**.** *Reading the data file*

Open with a text editor the data file `TenEvents.dat` *available on LEA.*

Note that every line starting with "#" is a comment. We will need to read only the data after all coments.

Our java code will have to read the pseudorapidity η, the azimuthal angle ϕ and the transverse momentum pt of each photon in a given event. Once these quantitites are extracted, your code should calculate the dipohoton invariant mass given by the formula

$$M^2 \quad = \quad 2\ |\mathbf{p}_T(1)||\mathbf{p}_T(2)|\left(\cosh\left(\eta_1-\eta_2\right)-\cos\left(\phi_1-\phi_2\right)\right) \tag{1}$$

For this, open the code `HiggsTaskA.java` *available on LEA. Use this code as a template to compute and print on the screen the diphoton invariant masses M associated to each of the ten events of the data file. If there are 3 photons in the same event, you need to calculate the invariant mass associated to each pair of photons (three values M_{12}, M_{13}, M_{23}).*

When you reach this point, show your results to a teacher.

Task (B)**.** *Constructing a histogram*

We will now move on to the real data file containing 500 000 events. Your group will be assigned one data file from the group of data files on LEA:
DiphotonsA.dat, DiphotonsB.dat, DiphotonsC.dat, DiphotonsD.dat, DiphotonsE.dat, DiphotonsF.dat, DiphotonsG.dat, DiphotonsH.dat or DiphotonsI.dat.

Now that we can calculate the diphoton invariant mass of each event, we want to create a histogram, counting how many diphoton events have a given invariant mass that lies within 2 GeV bins.

For example, how many diphoton events have invariant mass between 100 *GeV and* 102 *GeV.*

We want to create this histogram for masses between $90 GeV$ *and* 220 *GeV.*

We will need to create a counter for each 2 *GeV bin and increase its value everytime an invariant mass lies within it.*

For this, open the code `HiggsTaskB.java` *and use it to create the histogram out of the 500 000 events of your data file.*

Print the values on the screen and also send the results to a text file called "diphotonhistrogram.dat"

When you reach this point, show your results to a teacher.

Task (C)**.** *Statistical analysis - Significance*

- *Open the code* `HiggsTaskC.java` *in order to fit the data (interpolate data to a polynomial curve). The interpolated data is the "Background". The Difference between "Observed" and "Background" is the "Signal". The ratio* $\frac{Signal}{\sqrt{Background}}$ *is the "Signal significance" (this is just a poor man/woman's definition).*
- *Compute/print to file the signal significance.*
- *Conclude: if significance is 5 or more: discovery!*
- *at what invariant mass?*
- *Show your results in three graphs (use excel ONLY to plot):*
  1. *Diphoton events vs. Invariant mass*
  2. *(Observed - Background) vs. Invariant mass*
  3. *Signal Significance vs. Invariant mass*

When you reach this point, show your results to a teacher.

1 Appendix A

To read data we use the method "BufferedReader"

BufferedReader in = new BufferedReader(new FileReader(file));// This reads the file and puts it in a buffer

String x = in.readLine();// this reads a line of the buffer and converts it to a string "x"

String presstr1 = x.substring(0,1); // this reads the first character of the line "x" and calls it "pressstr1"

2 Appendix B

In order to create a histrogram we need to count elements in bins

to do so we round the calculated invariant mass to an even integer, by dividing it by two, rounding and then multiplying by 2.

EVEN = Math.round((Math.round(mass/2)*2));

3 Appendix C

To interpolate we will use the "PolynomialCurveFitter" method:

```
//Define the list "obs" of data to interpolate within the method
final WeightedObservedPoints obs = new WeightedObservedPoints();

for(int i=0; i<=10; i++)obs.add(5.0, 90+2*i, NM[90+2*i]);
// weight of 5 for low masses
for(int i=0; i<=20; i++)obs.add(1.0, 112+2*i, NM[112+2*i]);
// weight of 1 for intermediate masses
for(int i=0; i<=35; i++)obs.add(5.0, 154+2*i, NM[154+2*i]);
// weight of 5 for high masses

// Instantiate a third-degree polynomial fitter.
final PolynomialCurveFitter fitter = PolynomialCurveFitter.create(3);

// Retrieve fitted parameters (coefficients of the polynomial function).
// the fit is y = coeff[0] + coeff[1] x + coeff[2] x^2 + coeff[3] x^3
final double[] coeff = fitter.fit(obs.toList());

System.out.println(coeff[0]+" " +" " + coeff[1]+" " +" " + coeff[2]+ " " +" "+ coeff[3]);
```