Simulation Algorithms & Jet Algorithms

Ron Wu

3/22/14

Advisor: Professor Kyle Cranmer

Software References

[1] Bonnet *Introduction to MadGraph/MadEvent an informal tutorial*
[2] Degrande, Mattelaer *TASI 2013: FeynRules/MadGraph tutorial*
[3] MadEvent Minimal User Guide
[4] ROOTPrimer, ftp://root.cern.ch/root/doc/primer/ROOTPrimer.pdf

Simulation Algorithms

Madgraph is tool that generates tree-level (lowest perturbative order) matrix element for particles interactions. By specifying the initial, final states of all possible processes and inputs of tuning parameters not given by the SM, Madgraph will compute for any renormalizable, effective field theory beyond SM at universal FeynRules framework. Those generated matrix elements can be then used to compute cross section and decay width calculations. The main particle generator algorithm is Monte Carlo. It is based on the calculated matrix elements, i.e. probability, to randomly generate intermediate subparticles, which leads to many more different diagrams go to the same final state. The process of merging them and avoiding double counting is called parton showering, i.e. MLM Algorithm. The intricate merging process is the exact inverse process of jet-finding algorithm, which aims at separating jets.

References

[1] Alwall, *MadGraph 5 : Going Beyond* arXiv:1106.0522
[2] Alwall, Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions arXiv:0706.2569
[3] Krause Bachelor Thesis: The Impact of Dierent Monte Carlo Models on the Cross Section Measurement of Top-Pair Production at 7 TeV roton-Proton Collisions

[4] Visscher, MadGraph/MadEvent at work: from QCD to new physics

Jet-finding Algorithms

Jets are bunches of traces of remnant particles; they are reconstructed from energy depositions in calorimeter cells or from momenta of charged particles. In contrast to simulation algorithms, Jet algorithms are systematic ways of selecting useful jets to deduce information from the hadronic final states in high energy collisions. A good algorithm should accomplish the following (Snowmass requirement): 1) Infrared and collinear safety (renormalizable) The set of jets remain unchanged if soft emission or collinear splitting occurs. Because of non-perturbative effects, these things happen randomly. Also because in QCD calculation, these things correspond to divergent tree-level and its counterpart divergent loop matrix elements. 2) Adjustable to any order of QCD corrections. Hence high order terms should contribute small. 3) Contain estimation of small hadronization uncertainty. 4) Simple to implement in experiment and theoretical calculation.

Before 90's, e^+e^- collisions used JADE algorithm, which is a type of sequential recombination (clustering). Since the center of mass frame is stationary for e^+e^- collisions, the natural choice to separate particles is to consider the distance measure

$$y_{ij} = \frac{2E_iE_j(1-\cos\theta_{ij})}{Q^2}$$

where E_i the energy of the ith particle, and the difference of polar angles θ_{ij} of ith and jth particles. Q total energy of the event. If y_{ij} is below some cut off threshold, recombine the ith and jth particles into one particle. Repeat the process until y_{ij} is above the threshold for all i, j. One drawback of JADE is that soft gluons, often radiated far apart e.g. in reconstruction of W bosons and top quarks, may have a small E_iE_j, thus they get combined at very early stage. The remedy is to replace E_iE_j by $\min(E_i^2, E_j^2)$, and it is called $k_\perp$ algorithm. In this way, soft gluons are combined first with nearby high-order quarks.

One big difference between e^+e^- collisions and ep, $p\bar{p}$ hadron collisions is that most of energy in hardron collisions is not involved in hard reaction. So for hadron collisions 1) It's more convenient to analyze high $E_\perp$ cross section, which includes unobserved jets; 2) Distinguish clustering particles with small transverse momenta in the beam direction, because they do not undergone hard scattering; 3) Since the center of mass frame is not at rest, but it is invariant under boost along the beam direction. Therefore the natural choice for separation is in terms of longitudinal momentum p_z, transverse energy $E_\perp$, azimuthal angle ϕ, and rapidity $y = \frac{1}{2}\ln\frac{E+p_z}{E-p_z}$ or pseudorapidity $\eta = -\ln\frac{\tan\theta}{2}$. They make up iterative cone (IC) algorithm.

Take DØ as an example. It considers every calorimeter cell with energy above 1GeV as a seed cell. Take one seed particle, then combine all ith particle such that

$$\sqrt{(\eta_i - \eta_{seed})^2 + (\phi_i - \phi_{seed})^2} < 0.7$$

into a jet. Then repeat the process. Sorting particles lying in overlapping cones into a jet are dealt with extra rules, which lead to three different approaches. 1) progressive removal. Pick the seeds with largest p_z, then construct cone, then remove all particles in the cone; repeat the process. It is collinear unsafe. 2) split-merge. Construct cones using seeds with energy above 1GeV. In case of overlapping, if the ratio of maximum value of p_{shared} of shared particles over the maximum value of p of all particles in overlapping cones > 0.5, as an example for DØ, then merge overlapping cones into a jet. Otherwise split the shared particles to the closest overlapping cones. A variation of split-merge is split-drop, where non-shared particles in the overlapping cones are dropped. Both split-merge and split-drop are infrared unsafe.

References

[1] Chekanov, *Jet algorithms: a mini review* arXiv:hep-ph/0211298

[2] Salam *Towards Jetography* arXiv:0906.1833

[3] Ellis,Huston, *Jets in Hadron-Hadron Collisions* arXiv:0712.2447