Problem Set for Hadron Collider Physics 2015
Prof. Dr. Karl Jakobs, Dr. Karsten Köneke
Illustration of jet algorithms

This is NOT a problem set! We will rather discuss jet algorithms based on the examples shown below. These examples are designed to illustrate common issues with some jet algorithms and how others avoid those issues.

---

1. **Cone-jet algorithm**

   In the past, the most popular jet reconstruction algorithm at hadron colliders was a seeded-iterative-cone algorithm, despite the fact that such an algorithm is not "infrared-safe". The algorithm runs on calorimeter clusters as inputs (but a list of particles can also be given as inputs), and it proceeds as follows:

   (i) The clusters or particles above a certain threshold (say $p_T > 1\,\mathrm{GeV}$) are sorted in order of p_T starting from the highest to lowest transverse momentum.

   (ii) The algorithm starts with the highest transverse momentum cluster/particle which has not been assigned to a jet, and a temporary jet is formed containing all particles within a certain geometric distance to the seed particle. In this problem, we will use $\Delta R = \sqrt{(\Delta y)^2 + (\Delta\phi)^2} < 0.4$. The jet centroid (or barycenter) is also computed, which is the average p_T-weighted η and ϕ of the particles in the jet:

   $$\eta_{jet} = \frac{\sum_i^{clusters} p_{T,i} \cdot \eta_i}{\sum_i^{clusters} p_{T,i}} \quad \text{and} \quad \phi_{jet} = \frac{\sum_i^{clusters} p_{T,i} \cdot \phi_i}{\sum_i^{clusters} p_{T,i}}.$$

   (iii) From this new jet centroid, the algorithm is re-run, keeping all clusters/particles within $\Delta R < 0.4$, and obtaining a new jet centroid. The iteration continues until the jet centroid stabilizes, and a stable jet has been reconstructed. The jet momentum is then formed from the vector sum of the individual cluster momenta. Please note that throughout this and the next problem, you can simply approximate the vector sum of the jet momenta as a scalar sum of the jet p_T.

   (iv) The particles associated with the jet are removed from the list, and the algorithm starts again with the next highest transverse momentum cluster/particle still available. The algorithm stops when there are no more particles left.

   (v) There is also typically a merging/splitting procedure, which treats the overlap regions between two jets in a systematic way. Jets can be merged if the overlap region is large. For this Problem, we will ignore this step.

   (vi) Jets are also usually required to exceed a threshold in transverse momentum. For this problem, we will use $p_T > 10\,\mathrm{GeV}$.

   Run the cone algorithm described above on one hypothetical event by hand using the input particles (or clusters) listed in Table 1.

   (a) What jets finally result from the cone algorithm described above?

   (b) Now consider a slight change in the particle list. Particles 1 and 2 will suffer "collinear" splitting, a process that gives "infinite probability" when calculated in perturbative QCD. Thus, instead of particles 1 and 2, we consider particles 1a and 1b, as well as particles 2a and 2b, respectively, as given in Table 2. How does the jet content of the event change under this redefinition of particles?

Table 1: Particles (or calorimetric clusters) for one event.

Particle/Cluster Number	p_T [GeV]	y	ϕ
1	60	1.5	1.8
2	30	0.2	0.2
3	26	-0.1	0.4
4	25	0.4	-0.14
5	9	-0.15	0.45
6	8	0.5	-0.1
7	6	1.4	1.75

Table 2: New particles that replace particle 1 and 2 in Table 1.

Particle/Cluster Number	p_T [GeV]	y	ϕ
1a	55	1.5	1.8
1b	5	1.5	1.82
2a	20	0.23	0.17
2b	10	0.16	0.25

2. **(Anti-)k_{T}-jet algorithm**

Let's now consider "infrared-safe" algorithms, in particular the k_T and Anti-k_T algorithms, with parameter $R = 0.4$. One should note that both the ATLAS and CMS experiments have chosen to use the Anti-k_T algorithm as the standard jet reconstruction algorithm.

The k_T algorithm should proceed as follows:

(i) From the list of clusters/particles above a certain threshold (say $p_T > 1\,\mathrm{GeV}$), determine the following quantities (sometimes strangely referred to as "distances"):

- For every pair of particles i and j, determine d_{ij}, where

$$d_{ij} = \min(p_{T,i}^{2a}; p_{T,j}^{2a})\ \frac{\Delta R_{ij}^2}{R^2}$$

where $\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$ and $a = 1$.

- For every particle i, also determine d_{iB}, where

$$d_{iB} = p_{T,i}^{2a}\,,$$

again with $a = 1$.

(ii) Determine the smallest "distance" between all the d_{ij} and d_{iB}.

- If this smallest distance is a d_{ij}, then combine particles i and j into a single particle (by adding their 4-vectors), and go back to step (i). Treat jets/particles as massless for this problem.
- If this smallest distance is a d_{iB}, then call particle i a jet, and remove it from the list of particles. Then go back to step (i) for the remaining particles.

(iii) The algorithm stops, when there are no more particles to consider. Jets are then removed below a certain transverse momentum threshold, let's say 10 GeV for this problem.

The Anti-k_T algorithm is equivalent to the k_T algorithm except that now, $a = -1$ (note that the case with $a = 0$ is called the Cambridge-Aachen jet algorithm).

Run the algorithms described above on one hypothetical event by hand using the input particles (or clusters) listed in Table 1.

(a) What jets finally result from the k_T algorithm described above?

(b) What jets result from using the Anti-k_T algorithm instead?

(c) Repeat part (a) with the change in particle list: instead of particle 2 from Table 1, consider particles 2a, 2b from Table 2 instead.

(d) What do you notice about the types of particles that the k_T algorithm first focuses on, in comparison with the types of particles that the Anti-k_T algorithm first focuses on?