

A Data Association Nightmare: Algorithmic Approaches to Tracking my Red Solo Cup at a Party

The Chaz¹

¹College Freshmen, Cranberry-Lemon University, Pittsburgh, PA, USA, Major Undecided

Abstract

Dude, I'm at this kegger and no one remembered to bring a sharpie. I'll set down my drink next to someone else's, and I've got no idea which one's mine. This other dude came back and he didn't know either. Dude! Okay, so, I realized this is a data association issue, and we have to track these cups better. They look identical except for mine which I spike with some Jim Bean because I'm fancy. Which I guess still looks like everyone else's cup. Crap! I can keep track of where my drink is, but someone inevitably puts their cup next to mine or moves it because they think it's theirs or someone needs one some for-beer pong and is checking for an empty. We checked all the drawers in the kitchen twice and we can't find a sharpie anywhere. Pens don't work so the only way is to use a better data association algorithm.

Keywords: Drinking, Data Association, Multiple Solo Cup Tracking, Global Nearest Neighbor, Joint Probability Data Association, Mahalanobis, Cup Distribution, Cup Sensing, Semi-Greedy Algorithm, Multiple Hypothesis Tracker

1. Introduction

I know what you're wondering. How is it so hard to track a cup at a party? Cups don't move. Let me tell you, you've gotta see this party. It's insane. Totally crazy. We throw some wild parties here. We're also about three shots in each at the start of this thing to pre-game, so the Signal to Noise Ratio (SNR) is really bad. There are only a few surfaces to put our drinks on so it's not super easy to keep track of your cup.



Figure 1: The cups at this party bro, it's out of control!

To make things worse. We're trying to pick a new secret handshake that's two handed, so I keep having to put my drink down to practice and brain strom. Drinking someone else's

drink is a major party foul and there have already been a ton of infractions!

2. Probabilistic Cup Models

I'm not gonna lie, we might be able to observe a cup position Z , but it's not exactly accurate at this stage in the game. At about my current level inebriation of about three shots and two beers and whatever's still in my system from last night, I'm about at a Figure 2 [1].

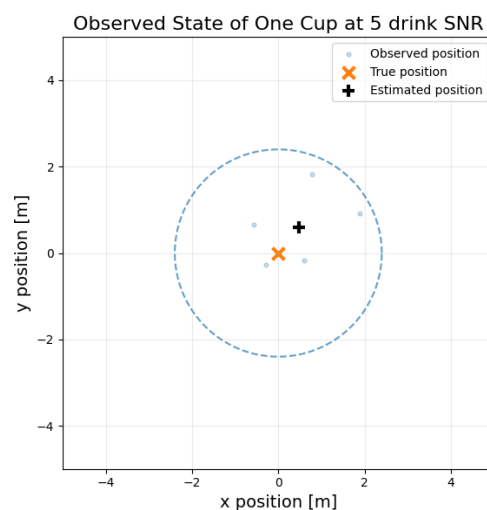


Figure 2: Low noise one cup estimate model

That's with my current alcohol induced SNR and one cup. In about an hour when I've had a few more, I'm gonna be at about a figure 3 [2].

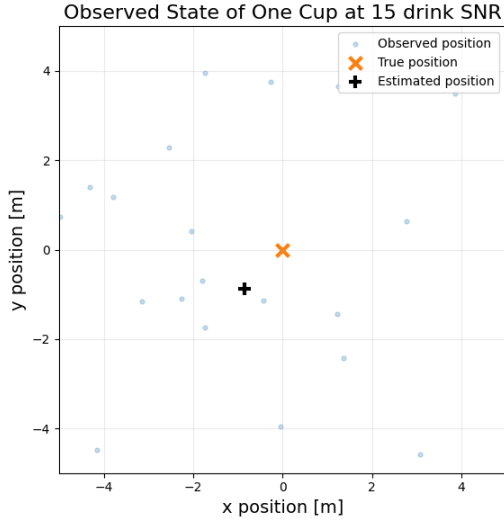


Figure 3: High noise one cup estimation

Now imagine throwing in three more cups in that state and you'll have figure 4.

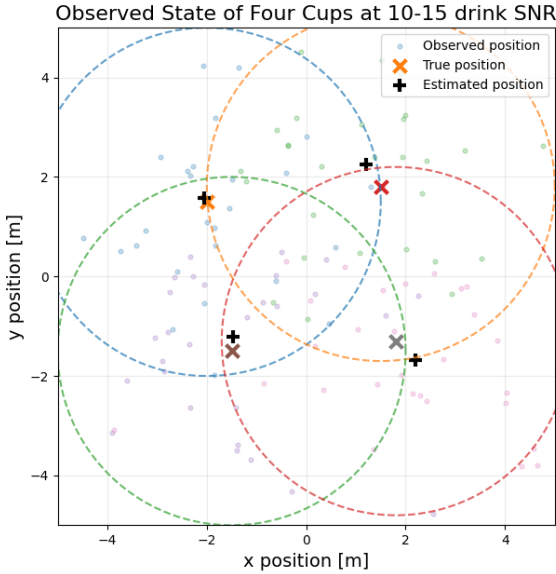


Figure 4: High noise four cup estimation model

We only have about five good tables that don't wobble too much for our drinks and at least a hundred people at this party. Some drinks can fit on the windowsill but let's be honest. We're heading for Figure 5 any moment now whenever Lauren comes back with the girls [3].

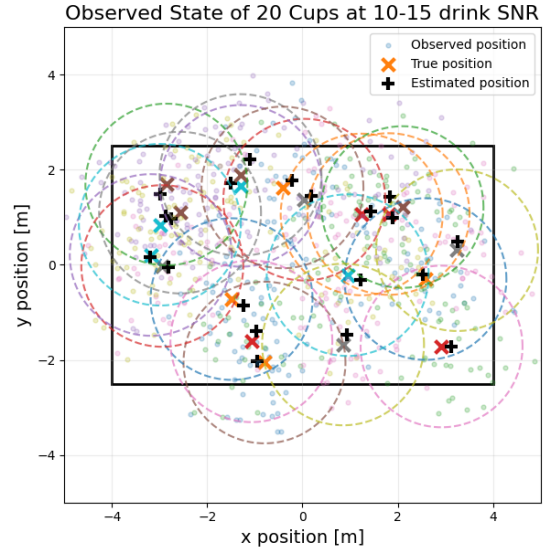


Figure 5: High noise party level cups estimation model

2. Multiple Cup Data Association

Okay so suppose we have N red solo cups belonging to P partiers. Now, only one cup can only be truly assigned to their true owner, but a lot of people are too worried and keep getting new cups because they can't figure out which cup is theirs. Pouring out beer is a worse party foul than drinking someone else's beer. The explosion of unattended beers at a party due to these two-party fouls is a result of Kyle's Paradox [4]. Let's just say $N \gg P$. The primary issue is figuring out which cup to associate with which partier. We're gonna try using the Global Nearest Neighbor (GNN), Joint Probability Data Association (JPDA), the Semi-Greedy, and the Multiple Hypothesis Tracker (MHT).

2.1 Global Nearest Neighbor

This one's pretty straight forward. We look at the total amount of partiers P , and we look at the cup observations on our fold out table Z , and we assign each cup to the closest partier in cost matrix C based on partier to cup distance. We construct all the hypotheses of the $Z \rightarrow P$ associations $\mathcal{H} = \{H_1, H_2, \dots, H_n\}$ for every possible hypothesis $H \in \mathcal{H}$. Next, while I'm in the bathroom, I'll calculate

$$J(H) = \sum_{(i,j) \in H} C_{ij}$$

And then pick my global minimum

$$H^* = \operatorname{argmin}_{H \in \mathcal{H}} J(H)$$

Hopefully I'll be able to do this before anyone moves a cup for the next frame.

2.2 Joint Probability Data Association

Some bros are sloshier than the others. Let's just say their cup positions are all over the place and that can be assigned a covariance with their beer induced SNR. We'll assume a normal distribution but there is nothing normal about this party. That will become our Cup Uncertainty Distribution (CUD), hah, CUD. Man, Todd is such a CUD [5]. Anyway, using the same notation as the GNN, our CUD likelihood of H is

$$L(H) = P(Z|H, P)$$

Where we can compute a posterior as

$$P(H|Z, P) \propto L(H)P(H)$$

And then we can calculate each partier to observation marginal probability

$$\beta_{i,j} = \sum_{H \ni (i,j)} P(H|Z, P)$$

Which lets us update each Partier cup estimate P using all the observations weighted by $\beta_{i,j}P^+$.

2.3 Semi-Greedy

In the semi-greedy algorithm, we'll make another cost matrix C and gate the cups that are WAAY too far from their owners. If their Mahalanobis distance is really far out, we set it as infinity. Next, we estimate the lowest cost valid pair in my immediate vicinity when finding a local cost minimum.

$$(i^*, j^*) = \operatorname{argmin}_{i \in P, j \in Z} C_{i,j}$$

Next, we remove P_{i^*} from P , remove Z_{j^*} from Z and move on to the next partier.

2.4 Multiple Hypothesis Tracking

Multiple Hypothesis Tracking (MHT) keeps track of each cup measurement and how they relate to the relative cost partier tracks in a set of forking paths. Over time I'll update the existing hypotheses $\mathcal{H}_{k-1}$ with cup observation Z_k to $\mathcal{H}_k$ before generating all associations between the partiers in H and observations Z_k which allows each feasible association A creating a new hypothesis:

$$H' = H \cup A$$

And then we can compute the hypothesis score using

$$S(H') = S(H) + \log P(Z_k|H')$$

After adding in H' to $\mathcal{H}_k$, we prune any hypothesis that's just totally wack. Like don't get mad at me if you're on the other side of the yard and leave you cup unattended bro. Then we pick the N_{max} best scoring hypothesis to merge back in.

3. Cup Database Maintenance

The cup association problem is only the first part of a red solo cup management algorithm. Since many cups go unattended and many appear out of nowhere as new people crash this shindig, we have to establish a good method for the life and death of our cup ownership database.

3.1 Cup Creation

When I'm looking at a table, I might have some issue seeing where cups are, but I can usually figure out how many cups are on a table. If I can create enough locally tracked cups of different cups that have not been associated to another partier for at least twenty seconds, I'd say that's a new cup and add that to the cup database.

3.2 Cup Tracking

This one's trickier than you'd think. One measurement would normally be enough but there's a lot of movement going on and a lot of people picking up and setting down their own drink. On average the cups don't move a lot but second to second, they're constantly moving. We basically have a single x-y coordinate for each cup in relation to the middle of the table. Every time I see the cup and the cup is associated with the previous cup state, I'll update that cup state Kalman filter with a generous amount of process noise in a no velocity model. This can get tricky with some of our house specific drinking games where people move a lot around the table like *Matt's Problem*, *Find your Shit*, *Strategic Chug Reserve*, or the most divisive game *Mutually Assured Intoxication* [6].

3.3 Cup Pruning

Now if someone's abandoned a drink, most likely because they forgot about it, we have to prune it from the list as a dead cup. Generally, if we don't see the cup move or get any emptier for about ten minutes; we call it a dead cup. These cups are tragically emptied by the Devan who either drinks em or knocks the table over when he gets the Ace of Spades when we're playing *Strategic Chug Reserve* [6]. That dude doesn't give a shit man.

4. Testing

I think I have about two hours before I am too far gone to remember any of this or be able to do any of these calculations so I'm gonna test each of these algorithms for thirty minutes each. Wish me luck boys!

5. Analysis and Discussion

Dude, that was a rough one and now I'm locked in the closet with Devan and the sock bag after all those infractions. I knew that was going to happen but it's BS man, everyone was drinking out of cups. They just got mad at me because I was writing it down and no one likes a snitch. Anyway, I tracked how many of my own cups I lost during each algorithmic test period and how many party fouls I committed in the process. The results are in table 1.

	Party Fouls	Lost Cups
GNN	37	3
JPDA	21	16
Greedy	13	2
MHT	6	1

Table 1: Party results

I didn't make a lot of cup related mistakes while using GNN, but I was so busy in the bathroom making calculations to brute force find the global optimum that I missed a lot of the party and kept some people out of the bathroom that ralphed all over the new carpet. I only lost three cups but I'm pretty sure Devan knocked em over while I was in the bathroom. Even then everything moved and all of the distances flattened out when we got a ton of cups.

JPDA was basically useless! I only picked up the right cup five times. I was slowing my drinking by then but even at a cup observation frame ever 60-90 seconds, all that beer induced SNR really spread out my covariances or whatever man. It sucked.

The Greedy algorithm almost worked. I kept my decisions local and was less worried about what was going on the other side of the yard. That didn't stop me from losing a couple of my beers when Joe-Joe's dog came ripping through and hopping up on the table. I don't know how he got that mutt approved as his service animal. The algorithm didn't stop me from drinking out of the wrong cup some, but I think we were just extra blitzed by then to make locally optimum desions well let alone globally optimum ones.

I'm not gonna lie, by the time I got to the MHT implementation, I was pretty slow to figure out anything. I waited and I watched. I looked carefully at the cups before deciding to drink. It helped me a lot dude, but with the amount of people that were at my table, I was experiencing a major hyp explosion merging all those branching trees in. By then, everyone lost it with me trying to track where everyone was putting their cups and I got put in the bad boy closet with Daren and the used sock bag.

5. Conclusion

Next time we gotta bring a sharpie, but man, that thing always goes missing. It's in high demand as soon as people

start crashing out on the couch. Someone's gotta draw dicks on faces and next thing you know, we have to send someone to pick up another sharpie pack at the CVS. Those things are like 8.99 now! I think the MHT and Greedy worked the best, but maybe there were some infractions I can't remember during that hour of the party.

References

- [1] I don't have time to run any sims right now so you're just gonna have to trust me bro with these plots :: Journal of Shut up Dude
- [2] If I remember last party, my drink was all OVER the place :: Annals of No-Duh
- [3] Okay, last time Lauren brought the women's rugby team over, we really threw down. This is about to get NUTS! :: Transactions of the Sexy Ladies Times
- [4] So Kyle is a legend on campus a realized that if no one pours out their beer but also can't remember which beer is there's and there's a significant threat of a paddling, let's just say the beer piles the FUCK up
- [5] Does this even need a reference? Everyone knows Todd is a CUD
- [6] *Delta Delta Delta's Big Book of Party Games* a Cranberry Lemon frat row staple.
- [7] Y. Bar-Shalom, P.K. Willett, and X. Tian, Tracking and Data Fusion: A Handbook of Algorithms, YBS Publishers, Storrs, CT 2011