**This is just a rough brainstorm**

1. Use principle components analysis to create PCs of the primary keys. We would somehow need to make these values all numeric or ordinal in order to take this approach, However, this would essentially create a series of loading where the higher loading represent the variables that had the highest degree of variability and therefore have potentially sparse identifiers which would better allow for identification of the tuple. We could then take these attributes and use them to create clusters
2. Cluster the attributes and give the user 2 tables in each data release: 1 with the cluster information, 2: table containing the remaining features of the dataset with an associated cluster and no aggregation. The user could then create a Cartesian join to the properties if more information was needed:
   1. The key to this approach is that the user would have to have a good amount of domain expertise

Very generalized example:

Clusters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cluster | Age range | zipcode region | m/f | frequent flier status |
| 1 | 0-18 | Midwest | 1 | 0 |
| 2 | 19-25 | Northeast | 0 | 0 |
| 3 | 26-38 | South Cental | 0 | A |
| 4 | 39-55 | South east | 1 | A-pref |
| 5 | 55+ | Desert mtn | 1 | A |

Data provided to users

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cluster | Orig | dest | dep\_date | Dep\_time | Business mkt? | # competitiors | Ontime? | NPS score | Itinerary type | total sked travel time | total actual travel time |
| 2 | DAL | MSY | 2/1/2016 | 6:00 | 0 | 2 | Y | 7 | Nonstop | 1:20 | 1:15 |
| 3 | OKC | HOU | 2/2/2016 | 7:00 | 1 | 2 | N | 2 | Nonstop | 1:45 | 2:15 |

These 2 tables could be joined to create a Cartesian product if the user wanted the anonymized attribute groupings to show up in the recordspace

1. Decision tree grouping. Not sure how this would work, but could use the principles of k-anonymity, l-diversity and t-closeness to determine the size of each node. Could be a really simple solution. Use clustering to group, then use decision trees/random forests to classify unknown observations?
2. Generalization
3. Suppression and managing the size of the release. In Netflix prize example they released a sparse data set that was less than the full data set and it still allowed for compromise of users identities and matching to other data sets. Obviously we can state that the release will be only an n% random sample of data, but it still leaves space for identification within

Quasi-Identifiers:

1. Standard Items
   1. Gender
   2. Zip Code of residence
   3. Date of birth
2. Home city. While this is related/analogous to zip code it’s another piece of information that could narrow the set of possible identifications

Sensitive Attributes:

1. Full flight history…as detailed in attacks there’s value in knowing how much one flyer flies annually.

Attacks:

Once we have the model in place, we can try some attacks.

Mixture of background knowledge and public data attack

1. I could pull Julie’s flight record in our anonymized dataset and we could calculate the probability that she was on one of 7 DAL-MSY flights on 2/23/2017 and link that to her facebook status and location changes? I’m sure she won’t mind ;)
2. Some ideas about attack purposes
   1. Industrial espionage…if a competitor is able to figure out my flight record, can they determine what cities and by extensions customers I’ve visited and met with?
   2. Personal safety…think someone vulernable to kidnapping and ransom attack
   3. Marriage and custody disputes…detecting a possible cheating spouse’s hidden flight history
   4. Other airlines to potentially find their high loyalty customers and find incentives to get them to fly more with them instead. If I can establish that customer A is flying my competitor 40% more than they fly me, there’s a value in getting their loyalty increased and reducing leakage. Other airline has access to some additional background information so this one’s harder.
3. Known flight legs/dates to extend into other flight history. If I meet someone on a specific flight, then how can I extend to know which other flights they have and more information about them?