Faceted Jostle: A Technical Description

A simple program in Jeeves will look like this:

```
let accuracy(D, m) = List.mean (List.tabulate (m (random_query D) ) len) in
let Logistic = Logistic_Classifier C1 in
let SVM = SVM_Classifier C2 in
label a in
    restrict a: lambda D. lambda m. accuracy(D, m) > 0.8
    let classifier = <a? Logistic : SVM> in
    (concretize classifier D) query
```

Any function in the `restrict` clause must have type $\mathcal{R} \to \mathcal{C} \to \texttt{bool}$, where $\mathcal{R}$ is the space of datasets and $\mathcal{C}$ is the space of models. Each of the faceted values will have type $\mathcal{R} \to \mathbb{R} \to \mathcal{C}$ where the first argument is the differential privacy constant. When `classifier` is concretized onto some dataset, it is expanded into the following code:

```
(concretize <a? Logistic : SVM>(D)) =
let (m1, e1) = search (S a) D Logistic in
let (m2, e2) = search (S a) D SVM in
if(e1 < e2) then m1 else m2
```

The expression `S a` retrieves any restrictions that have been set on a from our store, $\mathcal{S}$. Therefore, `search` has type

$$(\mathcal{R} \to \mathcal{C} \to \texttt{bool}) \to \mathcal{R} \to (\mathcal{R} \to \mathbb{R} \to \mathcal{C}) \to (\mathcal{C} \times \mathbb{R})$$

This makes it clear that `search` will return a real number representing the privacy of a model that satisfies the restriction along with the model. A natural way of implementing `search` is the following:

```
search S D M e = let model = M D e in
    if S model then (model, e) else search S D M (2*e)
search S D M = search S D M 0.001
```

Let the privacy of `search` be $f(\epsilon)$ where ϵ is the privacy of the returned model. We must have $f(\epsilon) \geq \epsilon$. There are some algorithms [1] where $f(\epsilon) = \epsilon$, but more work is needed to answer whether this is always possible. Let $\mathcal{D}_1, \mathcal{D}_2$ be the distribution of the model returned by `SVM` and `Logistic`, respectively. Then, the distribution of the returned model is

$$\Pr[\epsilon_1 > \epsilon_2]\mathcal{D}_1 + (1 - \Pr[\epsilon_1 > \epsilon_2])\mathcal{D}_2$$

Now, let $\mathcal{D}'_1$, $\mathcal{D}'_2$ and ϵ'_1, ϵ'_2 be the distributions and privacies returned for a database differing by 1 row. We have

$$\mathcal{D}'_1 \leq e^{\epsilon_1}\mathcal{D}_1 \quad \mathcal{D}_2 \leq e^{\epsilon_2}\mathcal{D}_2$$

it should not be higher than $\min\{e_1, e_2\}$ since this is the privacy usage of the best model.

References

[1] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Zhiwei Steven Wu. Accuracy first: Selecting a differential privacy level for accuracy-constrained ERM. *CoRR*, abs/1705.10829, 2017.