Minimal Schemas for a Category

Liang Ze Wong
wonglz@uw.edu
Agency for Science, Technology and Research (A*STAR), Singapore

When does a category have the structure of a database?

More precisely, given a category C, when is there a category S and a functor $F\colon S \to \mathbf{Set}$ such that C is the category of elements of F?

S is called a **schema** for C, and helps to structure the data of C.

There is always the trivial schema $S = C$, with F the constant functor sending each $x \in C$ to a point, but this does not help us understand the structure of C. Instead, we would like to find the smallest, or **minimal**, schema for C.

Let C be such that each coslice x/C has no non-trivial automorphisms.

The category C has the structure of a database where:

- **column headers are isomorphism classes of coslices in C**
- **entries in each column are objects with the same coslices**

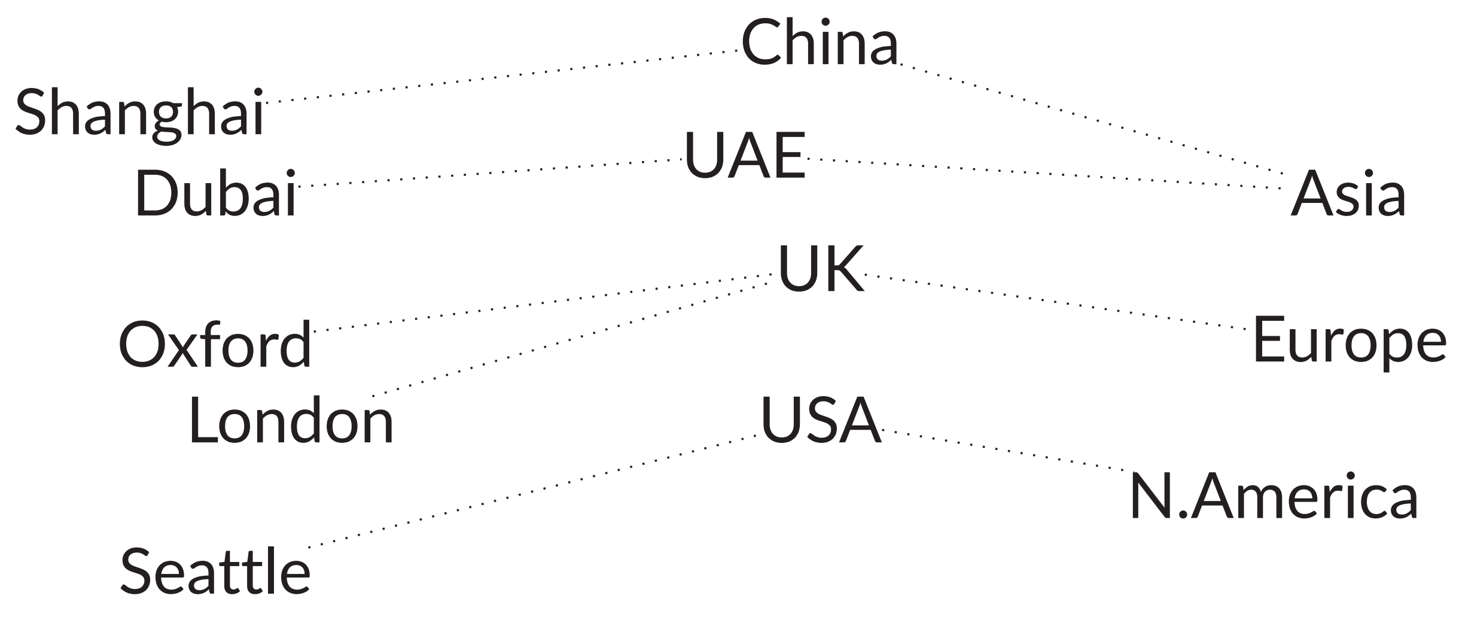
Equivalently, there is a schema S whose objects are isomorphism classes of coslices in C, and a functor $F\colon S \to \mathbf{Set}$ that sends the isomorphism class $[x/C]$ to the set of all $y \in C$ such that $y/C \cong x/C$.

Further, S is minimal in a precise sense: it is the terminal object in the category of schemas for S. This answers a question of Spivak (2014) for C satisfying the above hypothesis.

Does `word2vec` approximate a database?

The word embedding algorithm `word2vec` (Mikolov et al., 2014) takes a sample of sentences in a language, and embeds its words as points in a vector space.

`Word2vec` not only clusters similar words together, but also renders similar *relationships* between words as almost parallel difference vectors. For example:



In light of the similarity between the above diagram and the diagram for the category of elements, we speculate that:

- **The embedding given by `word2vec` approximates the structure of the category of elements of a database.** The objects of the schema S for this database are notions such as City, Country, Plural Noun, Singular Verb, etc. The morphisms of S capture relationships such as 'Is City in ...' or 'Is Plural of ...', and correspond to collections of almost parallel difference vectors in the embedding.
- **Words that are embedded close together have similar 'coslices'.** For instance, Oxford is similar to Shanghai, not because Oxford and Shanghai are both related to China in the same way, but because Oxford has 'its own China', namely UK.

We hope to make this analogy more precise, with an eye towards algorithms that *extract databases from unstructured data* (such as categories or collections of words).

Databases as Functors

Following Spivak (2012), a database is simply a functor $F\colon S \to \mathbf{Set}$. The category S is called the **schema** of the database. For example, the database

City	Country	Continent
Dubai	UAE	Asia
London	UK	Europe
Oxford	UK	Europe
Seattle	USA	N.America
Shanghai	China	Asia

may be expressed as a functor $F\colon S \to \mathbf{Set}$ where:

$$S = \begin{tikzcd} & \text{Country} \arrow[dr] & \\ \text{City} \arrow[ur] \arrow[rr] & & \text{Continent} \end{tikzcd}$$

$$\begin{aligned} F(\text{City}) &= \{\text{Dubai, London, Oxford, Seattle, Shanghai}\} \\ F(\text{Country}) &= \{\text{China, UAE, UK, USA}\} \\ F(\text{Continent}) &= \{\text{Asia, Europe, N.America}\} \end{aligned}$$

Note how objects of S appear as column headers for the database.

The Category of Elements

Any $F\colon S \to \mathbf{Set}$ has a **category of elements** denoted $\int F$. For the example above, the category of elements looks like:

$$\begin{tikzcd} & & \text{China} \arrow[ddr] & \\ & \text{Shanghai} \arrow[ur] \arrow[drr] & \text{UAE} \arrow[dr] & \\ & \text{Dubai} \arrow[ur] \arrow[rr] & & \text{Asia} \\ \int F \;= & \text{Oxford} \arrow[r] \arrow[drr] & \text{UK} \arrow[dr] & \\ & \text{London} \arrow[ur] \arrow[rr] & & \text{Europe} \\ & & \text{USA} \arrow[dr] & \\ & \text{Seattle} \arrow[ur] \arrow[rr] & & \text{N.America} \end{tikzcd}$$

Notice how the structure of $\int F$ is determined by the structure of S. The main question asks if a given category C is isomorphic to $\int F$ for some $F\colon S \to \mathbf{Set}$, i.e. if the structure of C is likewise determined by the structure of some S.

Coslice Categories

The **coslice category** x/C is the category of arrows $f : x \to y$ in C that start at x. Morphisms are commuting triangles:

$$\begin{tikzcd} & x \arrow[dl, "f"'] \arrow[dr, "g"] & \\ y \arrow[rr] & & z \end{tikzcd}$$

In the example of $\int F$ above, note that we have coslice isomorphisms

$$\text{Shanghai}/\textstyle\int F \;\cong\; \text{Oxford}/\textstyle\int F$$

even though Shanghai and Oxford are not isomorphic in $\int F$. We claim that this is the notion of similarity that `word2vec` captures.

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.

[2] David Spivak. Functorial data migration. *Information and Computation*, 217:31-51, 2012.

[3] David Spivak. Does there exist a terminal surjective discrete fibration out of C? *MathOverflow:157519*, 2014.