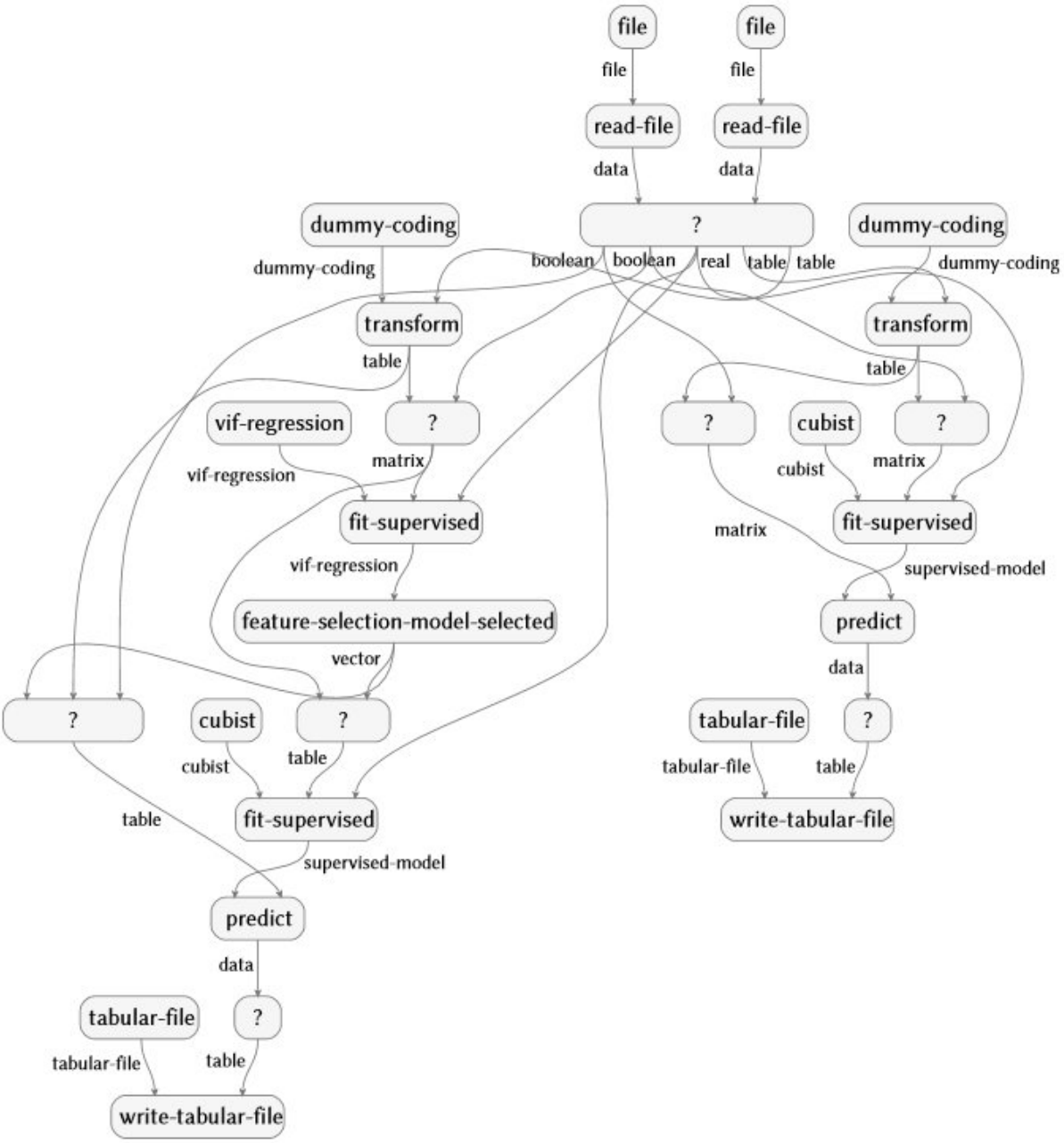
AI for code encourages collaborative, open scientific discovery

August 16 2018, by Kush Varshney



Semantic flow graph representation produced automatically from an analysis of rheumatoid arthritis data. Credit: IBM

We have seen significant recent progress in pattern analysis and machine intelligence applied to images, audio and video signals, and natural language text, but not as much applied to another artifact produced by people: computer program source code. In a paper to be presented at the FEED Workshop at KDD 2018, we showcase a system that makes progress towards the semantic analysis of code. By doing so, we provide the foundation for machines to truly reason about program code and learn from it.

The work, also recently demonstrated at IJCAI 2018, is conceived and led by IBM Science for Social Good fellow Evan Patterson and focuses specifically on data science software. Data science programs are a special kind of computer code, often fairly short, but full of semantically rich content that specifies a sequence of data transformation, analysis, modeling, and interpretation operations. Our technique executes a data analysis (imagine an R or Python script) and captures all of the functions that are called in the analysis. It then connects those functions to a data science ontology we have created, performs several simplification steps, and produces a semantic flow graph representation of the program. As an example, the flow graph below is produced automatically from an analysis of rheumatoid arthritis data.

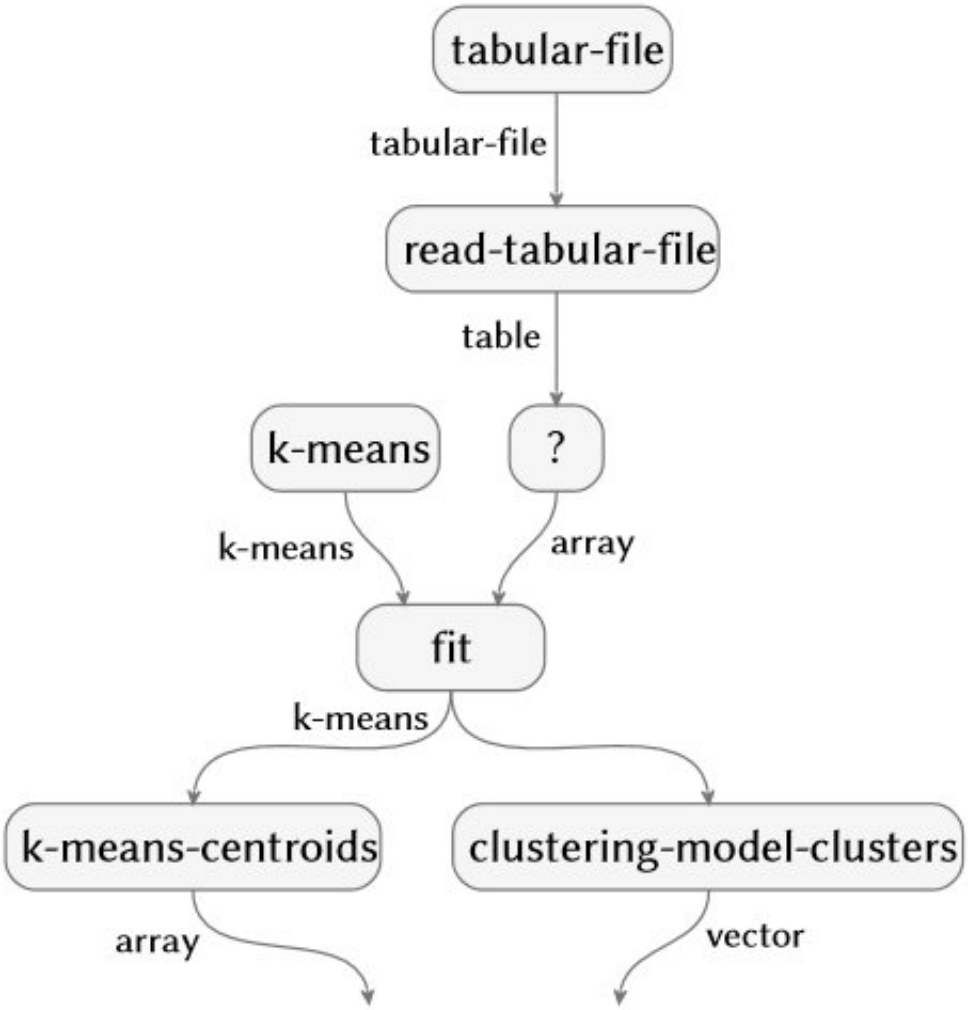
The technique is applicable across choices of programming language and package. The three code snippets below are written in R, Python with the NumPy and SciPy packages, and Python with the Pandas and Scikit-learn packages. All produce exactly the same semantic flow graph.

```r
iris = read.csv('iris.csv', stringsAsFactors=FALSE)
iris = iris[, names(iris) != 'Species']

km = kmeans(iris, 3)
centroids = km$centers
clusters = km$cluster
```

```python
import numpy as np
from scipy.cluster.vq import kmeans2

iris = np.genfromtxt('iris.csv',
    dtype='f8',
    delimiter=',',
    skip_header=1,
)
iris = np.delete(iris, 4, axis=1)

centroids, clusters = kmeans2(iris, 3)
```

```python
import pandas as pd
from sklearn.cluster import KMeans

iris = pd.read_csv('iris.csv')
iris = iris.drop('Species', 1)

kmeans = KMeans(n_clusters=3)
kmeans.fit(iris.values)
centroids = kmeans.cluster_centers_
clusters = kmeans.labels_
```

Credit: IBM



Credit: IBM

We can think of the semantic flow graph we extract as a single data point, just like an image or a paragraph of text, on which to perform further higher-level tasks. With the representation we have developed, we can enable several useful functionalities for practicing data scientists, including intelligent search and auto-completion of analyses, recommendation of similar or complementary analyses, visualization of

the space of all analyses conducted on a particular problem or dataset, translation or style transfer, and even machine generation of novel data analyses (i.e. computational creativity)—all predicated on the truly semantic understanding of what the code does.

The Data Science Ontology is written in a new ontology language we have developed named Monoidal Ontology and Computing Language (Monocl). This line of work was initiated in 2016 in partnership with the Accelerated Cure Project for Multiple Sclerosis.

More information: E. Patterson et al. Dataflow representation of data analyses: Toward a platform for collaborative data science, *IBM Journal of Research and Development* (2017). DOI: 10.1147/JRD.2017.2736278

This story is republished courtesy of IBM Research. Read the original story here.

Provided by IBM

Citation: AI for code encourages collaborative, open scientific discovery (2018, August 16) retrieved 27 April 2024 from https://phys.org/news/2018-08-ai-code-collaborative-scientific-discovery.html