Applied category theory in data science

Evan Patterson
Statistics Department, Stanford University

Ioana Baldini, Aleksandra Mojsilovic, Kush R. Varshney
IBM Research AI

Applied Category Theory: Bridging Theory & Practice
March 15-16, 2018, NIST, Gaithersburg, MD

Problem: Program understanding for data science

A classic, hard problem in AI:

> *How can we teach machines to* understand *code, as opposed to merely executing it?*

We try to solve an incarnation of this problem in data science:

> *How about understanding data analysis (in the form of computer code)?*

Our goals

Technical. Form semantic representations of data analyses capturing:

- generic concepts of computing (functions, programs, etc.)
- domain-specific concepts of data science (data, models, prediction, etc.)

Applications. Create an AI assistant for data scientists:

- Summarize data analyses in natural language text
- Discover related data sets or data analyses
- Automate meta-analysis and meta-learning

Two examples

Example 1: k-means clustering using NumPy and SciPy

In [1]:

```
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)
```

Example 2: k-means clustering using pandas and scikit-learn

In [2]:

```
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_
```

Demo: Semantic flow graphs

Example 1: k-means clustering using SciPy
Example 2: k-means clustering using scikit-learn

The two examples have the **same** semantic representation.

demo / scipy_clustering_kmeans

k-means clustering on Iris dataset using NumPy and SciPy

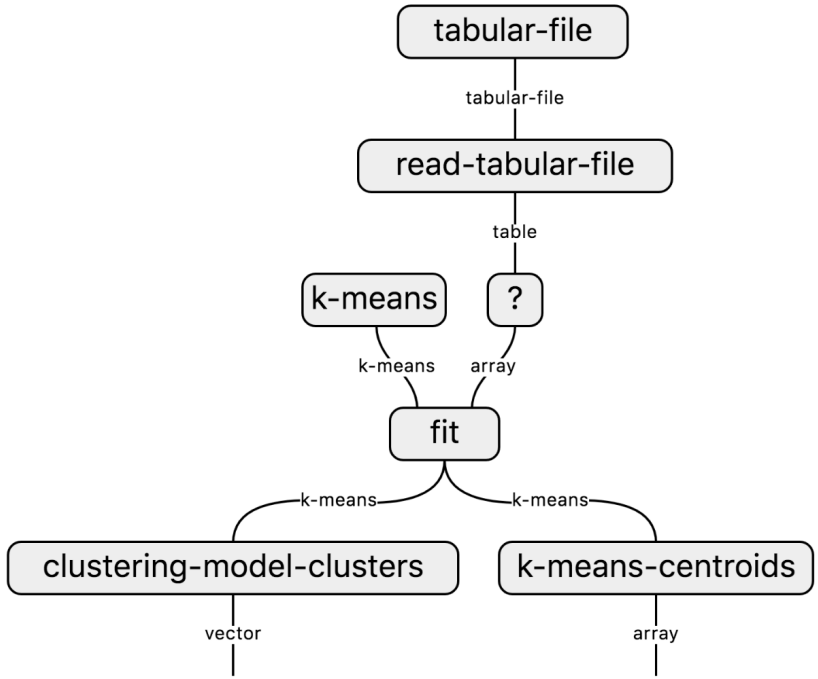
We run a k-means clustering analysis on the Iris dataset using the scientific computing libraries NumPy and SciPy. There are three species of flowers, so we use three clusters. Even with this knowledge, the Iris data is known to not cluster correctly using k-means clustering.

Copy Save Python

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

iris = np.genfromtxt('opendisc/opendisc/integration_tests/data/datasets
iris = np.delete(iris, 4, axis=1)

centroids, clusters = kmeans2(iris, 3)

```

Raw flow | Semantic flow

Labels Export



demo / sklearn_clustering_kmeans

k-means clustering on Iris dataset using pandas and sklearn

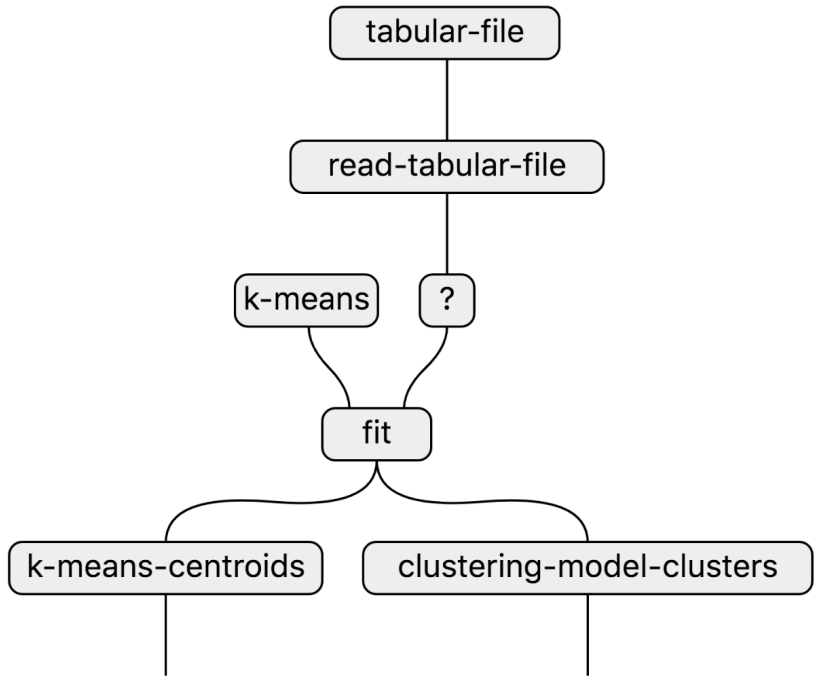
We run a k-means clustering analysis on the Iris dataset using the data science libraries pandas and scikit-learn. There are three species of flowers, so we use three clusters. Even with this knowledge, the Iris data is known to not cluster correctly using k-means clustering.

Copy Save Python

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

iris = pd.read_csv('opendisc/opendisc/integration_tests/data/datasets/i
iris = iris.drop('Species', 1)

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

```

Raw flow | Semantic flow

Labels Export



Methodology

Derive semantic representation from computer program in two steps:

1. Program analysis
   Transform computer program to dataflow graph (raw flow graph)
2. Semantic enrichment
   Add semantic content to dataflow graph (semantic flow graph)

Semantic enrichment is

- mainly where the applied category theory happens
- based on an ontology about data science

Concepts as category

Concepts record abstract ideas from data science, generating a monoidal category with

1. Types (objects): data tables, statistical models, etc.
2. Functions (morphisms): reading data from a file, fitting a model, etc.

Relevant doctrines:

- cartesian category: minimal model of deterministic computation
- cartesian closed category: adds function types and lambda abstraction
- traced cartesian category: adds looping/recursion

Annotations as functor

Annotations map code in data science libraries to concepts in the ontology:

$$\text{Code category } \mathcal{C}_0 \xrightarrow{\text{Annotations}} \text{Concept category } \mathcal{C}$$

It extends to the category of elements:

$$\text{Raw flow graphs} = [\mathcal{C}_0, \mathbf{Set}] \xrightarrow{\text{Annotations}} [\mathcal{C}, \mathbf{Set}] = \text{Semantic flow graphs}$$

Caveat: These are partial mappings.

Demo: Data Science Ontology

Annotations from Data Science Ontology:

- k-means clustering in SciPy
- k-means clustering in scikit-learn

demo / scipy_clustering_kmeans

k-means clustering on Iris dataset using NumPy and SciPy

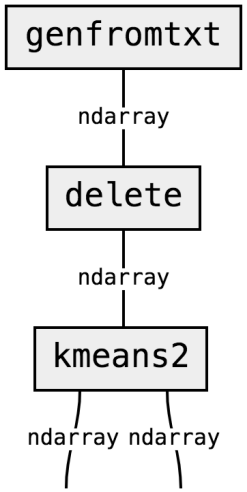
We run a k-means clustering analysis on the Iris dataset using the scientific computing libraries NumPy and SciPy. There are three species of flowers, so we use three clusters. Even with this knowledge, the Iris data is known to not cluster correctly using k-means clustering.

Copy Save Python

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

iris = np.genfromtxt('opendisc/opendisc/integration_tests/data/datasets
iris = np.delete(iris, 4, axis=1)

centroids, clusters = kmeans2(iris, 3)

```

Raw flow | Semantic flow

Labels Export



Annotation k-means clustering in scipy

Language Python

Package scipy (PyPI)

ID kmeans2

Kind → function

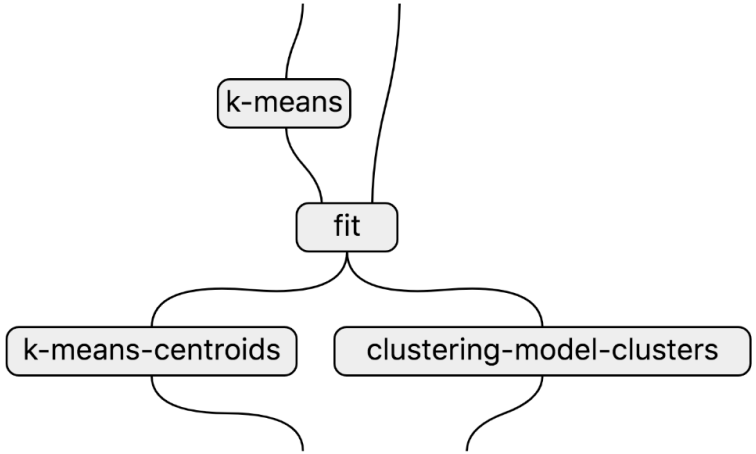
Name k-means clustering in scipy

Python function `scipy.cluster.vq.kmeans2`

Input
```
1
0
```

Output
```
__return__.0
__return__.1
```

Definition
- compose
  - product
    - construct
      - pair
        - k-means
        - clustering-model-n-clusters
    - id
      - array
  - Function
    - fit
    - product
      - k-means
      - data
    - k-means
  - pair
    - k-means



Thank you!

Further reading

Previous paper: Patterson et al, 2017. Dataflow representation of data analyses: Towards a platform for collaborative data science.

Forthcoming paper: Patterson et al, 2018. An ontology language, ontology, and algorithm for semantic representation of computer programs.