Clustering by contrast

Cyril CHHUN

Télécom Paris

Advisor: Jean-Louis DESSALLES

June 20, 2019

Outline

Introduction

What are the end goals of contrast learning?

Design a **clustering algorithm** able to:

- understand the meaning of “small bacteria” and “small galaxy” without going through the set of “small” objects
- produce relevant descriptions: “it’s a singer who has ten million views on Youtube”
- produce negations and explanations: “she is *not* a writer”
- detect anomalies: a talking cat
- learn from a single example: a “Siamese cat”

Impossibility theorem

Which properties would we expect of a clustering algorithm?

- Scale-invariance, richness, consistency
- Kleinberg (2002)[1]: it is impossible to design a distance function-based clustering algorithm which verifies those three properties.
- Solution: forsake one of those properties or use non-metric functions

[1] Jon Kleinberg. An impossibility theorem for clustering. *Advances in Neural Information Processing Systems 15*, 2002.

Vocabulary

- Object: observed instance
- Prototype: mental representation of a group as a basic object
- Contrast: “difference” between two objects
- Weight: number of times a prototype has been recalled
- Deviation: acceptable range of an object’s properties compared to its prototype
- Order: real-life observations are first-order objects, contrasts are second-order objects, etc.

Design

Prototypes

How to represent prototypes?

$$\begin{array}{ccc} \text{Mean} & \text{Deviation} & \text{Weight} \\ \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_m \end{bmatrix} & \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_m \end{bmatrix} & w \end{array}$$

Design

Finding the clusters

Given object b, how to find the best prototype a of deviation a'?

- Dimension-agnostic, scale-invariant, not density-based.
- The prametric function

$$d(a, b) \mapsto \sum_{j=1}^{m} \mathbb{1}\left(\frac{|a_j - b_j|}{a'_j} > \theta_j\right)$$

  verifies **scale-invariance along any axis**.
- It is not a distance, as none of the three properties are verified!
- Problem: many prototypes can verify the smallest distance.

Design
Comparing the clusters

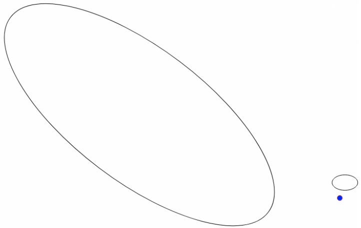

Figure: The smaller cluster seems more reasonable: how to avoid the hub?

- We simply take the best prototypes two by two and choose the one whose mean is closer to the object along the most dimensions. The other cluster is eliminated.
- Using this rule, we make a tournament and pick the winner.
- Deviations are not used in this step so as to avoid hubs.

Design

Updating the memory

How to stock the new information in the memory?

- The object is added as a prototype no matter what, with a deviation equal to ε times itself and a weight of 1
- The winning cluster is updated as follows:

$$\begin{aligned} mean &= \frac{weight * prototype + object}{weight + 1} \\ deviation &= \frac{weight * deviation + |prototype - object|}{weight + 1} \\ weight &= weight + 1 \end{aligned}$$

- We enforce a limited memory to cope with initial errors and improve efficiency. Unused prototypes are forgotten first.

Design

Skeleton

```python
def feed_data_online(data):
    for obj in data:
        closest_clusters = find_closest_clusters(obj)
        winner = cluster_battles(obj, closest_clusters)
        update_memory(obj, winner)
```

- Clustering: simple loop with complexity $O(mem\_size \times n)$
  $\rightarrow$ Online learning

Understanding results

- Softer clustering than k-means; different ways to classify when seeing a new object:
  - Assign object b to prototype a if $d(a, b) = 0$
  - Find the closest prototype to the object (by tournament for example)

Live demonstration

What about contrasts?

How to extract relevant contrasts?

- The contrast features should be meaningful, i.e. low-dimensional and applicable between similar objects.
- Given an object b and its closest prototype a, we extract the contrast c such that

$$c_j = (a_j - b_j) \cdot \mathbb{1}\left(\frac{|a_j - b_j|}{a'_j} > \theta_j\right)$$

- Example: seeing a black tomato would give a "red-to-black" contrast.

What about contrasts?

How to stock the contrasts in memory?

- We can use the same principle! Contrast-prototypes with mean, deviation and weight.

Then, how to refine the contrasts?

- We can use the **same procedure**!

Second demonstration

Feedback on the checklist

- ✓ understand the meaning of "small bacteria" and "small galaxy" without going through the set of "small" objects
- ✗ produce relevant descriptions: "it's a singer who has ten million views on Youtube"
- ✗ produce negations and explanations: "she is *not* a writer"
- ✓ detect anomalies: a talking cat
- ✓ learn from a single example: a "Siamese cat"

Conclusion

- The algorithm is dimension-agnostic and verifies scale-invariance
- It learns on-the-fly and has a reasonable complexity (linear on average)
- Designed to be used on relatively high-level datasets
- Contrasts still need testing: some inconsistent results can appear