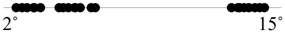
Supervising Unsupervised Learning

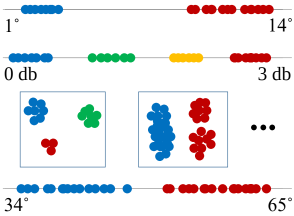
Vikas K. Garg & Adam Kalai

Clustering problem in isolation:



How many clusters?

Clustering repository:



Contributions

- Introduce a principled framework to evaluate unsupervised settings
- Show how to transfer knowledge across heterogeneous datasets
  - different sizes, dimensions, representations, domains...
- Design provably efficient algorithms
  - select clustering algorithm and number of clusters,
  - determine threshold in single-linkage clustering
  - remove outliers, recycle problems
- Make good *meta-clustering* possible
  - introduce *meta-scale-invariance* property
  - show how to circumvent Kleinberg's impossibility result
- Automate deep feature learning across very small datasets
  - encode diverse small data effectively into big data
  - perform non-trivial zero shot learning

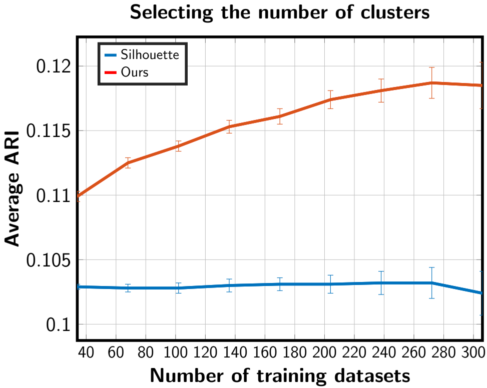
General approach

- Define a meta-distribution μ over all problems in the universe
- Each training sample is a *dataset* drawn i.i.d. from μ
- Learn a mapping from an *intrinsic* measure to an *extrinsic* measure
- Intrinsic measure avoids labels and abstracts away heterogeneity
- Each test problem is drawn from μ but labels are hidden
- Compute intrinsic measure on test and predict the extrinsic quality
- Encode covariance of small datasets for deep zero-shot learning

Number of clusters

Summary

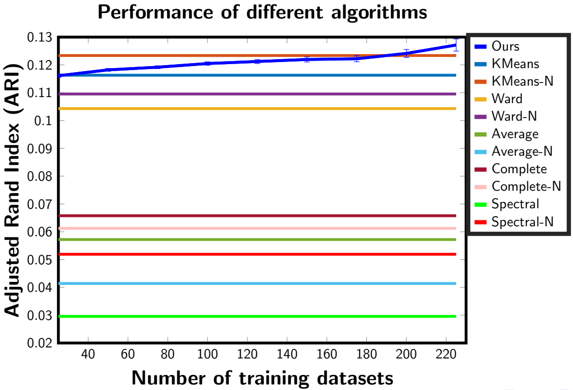
Run *k*-means algorithm with different *k* on each train dataset.
Use Silhouette Index (SI) as intrinsic measure.
Use Adjusted Rand Index (ARI) as extrinsic measure.



Clustering algorithm (assume fixed k for simplicity)

Summary

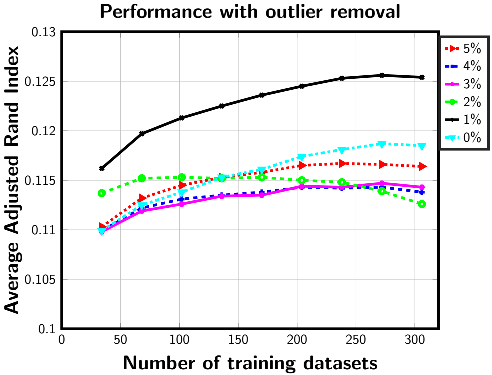
Run different algorithms to get k clusters & compute SI.
Form a feature vector from SI and dataset specific features (e.g. max and min singular values, size, dimensionality).
Use Adjusted Rand Index (ARI) as extrinsic measure.



Fraction of outliers

Summary

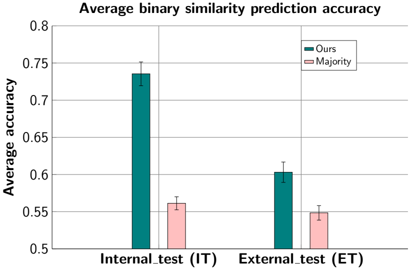
Remove points with large norms, cluster other points, and compute SI.
Put the removed points into clusters, and compute ARI.
Find the candidate fraction that performs best on test set.
Extensions possible to customize fractions for each test set.



Deep learning binary similarity function

Summary

Sample pairs of examples from each small dataset.
For each pair, also include covariance features specific to its dataset.
Label 1 if the sampled pair comes from same cluster, 0 otherwise.
Train a deep net classifier on all the pairs together.
Predict whether test pair comes from same cluster or not.



See you... ☺

Tue Dec 4th 05:00 – 07:00 PM
Room 210 & 230 AB
Poster #164