universität wien

RandomLink - Avoiding Linkage-Effects by emplyoing Random Effects for Clustering

---

DEXA 2020, Data Mining - Session 5

Gert Sluiter [1] Benjamin Schelling [1] Claudia Plant [1,2]

[1]University of Vienna | [2]ds:UniVie

15. Sep. 2020

Clustering Example

Clustering of Image Features

Problem: groups in unlabeled data

- Image features $M_{1000 \times 500}$ from pretrained Convolutional Neural Network
- PCA reduced dimensionality for visualization
- Question: are there any groups?

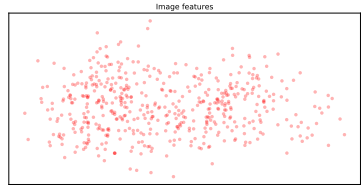

Figure: image features from 500 instances mapped into 2D Plane

Clustering Example

Clustering of Image Features

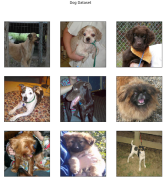

Figure: Dogs from Cats and Dog Dataset [1]

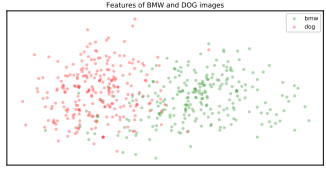

Figure: Color coded Groundtruth

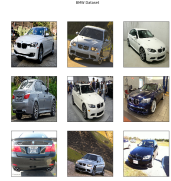

Figure: Cars from BMW Dataset [2]

Insights

- Clustering showed groups
- CNN Model that generated the features has learnt meaningful filters
- Clustering does work in high dimensional feature space

Clustering Example

Clustering of Image Features

Figure: k-Means clustering $k = 2$

Figure: k-Means clustering $k = 4$

Figure: DBSCAN Clustering

- **Problems:** Result depends on algorithm and parameters
  - How to find good algorithm
  - Is the clustering applicable for the data (distribution, noise)
  - How to select parameters

RandomLink Clustering

The Idea

Fundamental Assumption

Let the affinity be defined by the pairwise similarity, deleting edges randomly using their weight as probability of being deleted leads to a useful clustering at some point which is not constraint to certain shapes of clusters.

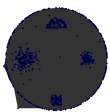

Figure: all points linked

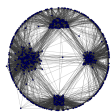

Figure: still one component

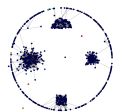

Figure: start of decomposition

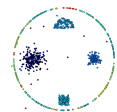

Figure: state of peak NMI

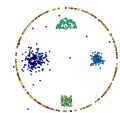

Figure: clustering gets worse if too many links removed

Related Work

Hierarchical Clustering

Single Link Clustering: [3, 4, 5, 6]

Differ in how they measure distance between clusters, Agglomerative or Divisive

- Input: Data ⟶ Distance Matrix
- Routine: iteratively merge or split clusters that are the closests according the distance used and recompute the distances until k clusters or a threshold is reached. Returns a tree like hierarchy of clusters
- Drawbacks: Single Link effect [7], sensitive to outliers and noise, Parameter choice

Related Work

Graph-based Clustering

Highly Connected Subgraphs (HCS) [8]

- Input: Thresholded Graph
- Routine: Recursive minimum-cut [9] to split graph(s) into sub-graphs until sub-graphs are highly connected i.e. $\frac{|V|}{2}$ edge deletions needed for further splitting
- Drawbacks: parameters, mincuts can be costly and single data points tend to be isolated

Related Work

Graph-based Clustering

Chameleon [10]

- Input: kNN Graph
- Routine: split graph into many sub-graphs with repeated min-cuts [11], then join the sub-graphs as long as the relative interconnectivity and the relative closeness are satisfied
- Drawbacks: 3 Parameters, even though their influence should be small according to the author, costly min-cut

Relative Interconnectivity

$$RI(C_i, C_j) = \frac{|EC_{C_i,C_j}|}{\frac{|EC_{C_i}|+|EC_{C_j}|}{2}}$$

Relative Closeness

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{(C_i,C_j)}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{(C_i)}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{(C_j)}}}$$

RandomLink Clustering

Implementation

Algorithm 1 RandomLink

Require: Data D

procedure RandomLink(D)

$S \leftarrow$ similarity matrix(D)

$sum_S \leftarrow \sum(S)$, $score_{max} \leftarrow 0$

return *Labels*

end procedure

Similarity Matrix

Fully-connected Dataset

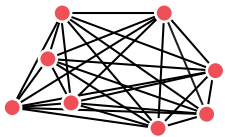

Figure: Fully-connected Graph

- Similarity Matrix $S_{n \times n}$ for $D_{d \times n}$

Probability of Link e_x being deleted

$$p(e_x) = \frac{I(e_x)}{\sum_e I(e)}$$

RandomLink Clustering

Implementation

Algorithm 1 RandomLink

Require: Data D
  procedure RandomLink(D)
    $S \leftarrow$ similarity matrix(D)
    $sum_S \leftarrow \sum(S)$, $score_{max} \leftarrow 0$
    while $sum_S > 0$ **do**
      delete random link e
      $sum_S \leftarrow sum_S$-length(e)
      set index of e 0 in S
      stack.push(index of e)
    end while

    return *Labels*
  end procedure

Roulette Wheel Selection [12]

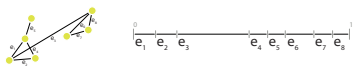

Figure: Dataset before Link removal

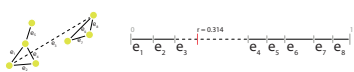

Figure: Random value $r = 0.314$ points to e_3

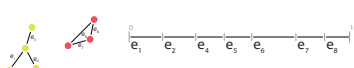

Figure: Dataset with e_3 deleted

RandomLink Clustering

Implementation

Algorithm 1 RandomLink

Require: Data D

procedure RandomLink(D)
- $S \leftarrow$ similarity matrix(D)
- $sum_S \leftarrow \sum(S)$, $score_{max} \leftarrow 0$
- **while** $sum_S > 0$ **do**
  - delete random link e
  - $sum_S \leftarrow sum_S$-length(e)
  - set index of e 0 in S
  - stack.push(index of e)
- **end while**
- $ds \leftarrow$ Disjoint Set.make_set(S)
- **while** ds.n_connected_components > 1 **do**
  - ds.union(stack.pop())
  - **if** (ds.n_connected_components changed) **then**
    - $score \leftarrow$ stopping criterion(ds.connected_components, D)
  - **end if**
  - **if** ($score > max_{score}$) **then**
    - $Labels \leftarrow ds$.connected_components, $score_{max} \leftarrow score$
  - **end if**
- **end while**
- **return** *Labels*

end procedure

Disjoint-Set / Union-Find Datastructure

- Init: initialized after links have been deleted
- Union: With links from stack
- Find: For connected components analysis
- Optimization: Path Compression and Rank by Size?

RandomLink Clustering

Implementation

Algorithm 2 Stopping Criterion

Require: connected components *cc*, Data *D*

procedure evaluate stopping criterion(*cc*,*D*)

Label ← k-Means (*D*,k=|*cc*|)

return NMI (*Label*,*cc*)

end procedure

Normalized Mutual Information

- Symmetric, $NMI(X, Y) = NMI(Y, X)$
- Score between 0 and 1
- Independent of permutations from class labels
- Different normalizations (geometric mean used)

Definitions

$$NMI(X, Y) = \frac{2I(X, Y)}{\sqrt{H(X) \cdot H(Y)}} \qquad I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \qquad H(X) = -\sum_{x \in X} p(x) \log p(x)$$

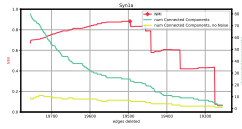

Figure: NMI history for deleting last remaining links

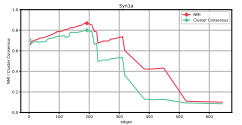

Figure: NMI and Stopping Criterion Score for one Run

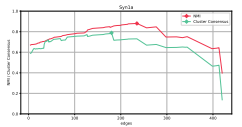

Figure: NMI and Stopping Criterion Score for another Run

RandomLink Clustering

Adding or Deleting Links

Reduce operations with Bottom-Up Edge Insertions?

- Single Edge connecting two Clusters leads to worse results

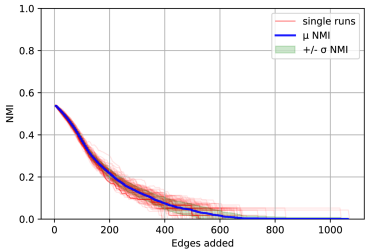

Figure: Bottom-Up Edge Addition

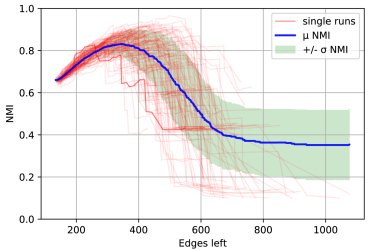

Figure: Top-Down Edge Deletion

RandomLink Clustering

Saving Time

Better Solution Bottom-Up and Top-Down combined

- Create order of edge deletion first (Top-Down)
- Add edges in reversed order which is defined in step 1 (Bottom-Up)

Benefits

- Reduced Stopping criterion and connected components calculations as the Graph generally will be fully connected for a fraction of the total amount of edges added
- Fixed order, parallelization, no dependencies between states

RandomLink Clustering

Runtime

Algorithm 1 RandomLink

```
Require: Data D
    procedure RandomLink(D)
        S ← similarity matrix(D)
        sum_S ← ∑(S), score_max ← 0
        while sum_S > 0 do
            delete random link e
            sum_S ← sum_S-length(e)
            set index of e 0 in S
            stack.push(index of e)
        end while
        ds ← Disjoint Set.make_set(S)
        while ds.n_connected_components > 1 do
            ds.union(stack.pop())
            if (ds.n_connected_components changed) then
                score ← stopping criterion(ds.connected_components, D)
            end if
            if (score > max_score) then
                Labels ← ds.connected_components, score_max ← score
            end if
        end while
        return Labels
    end procedure
```

Runtime complexity for every Routine

Routine	**Complexity**
Similarity Matrix	$O(n^2)$
Link Deletion	$O(n^2 \cdot log_2(n))$
Disjoint-Set make	$O(n)$
Disjoint-Set update	$O(\alpha(n) \cdot n^2)$
Stopping Criterion	$O(n \cdot n)$

biggest Term: $O(n^2 \cdot log_2(n))$

Evaluation

Stopping Criterion Evaluation

Cluster Count k

- reduces runtime
- introduces parameter
- produces slightly better results for real world data

Stopping Criterion

- no parameters
- slightly worse results

	Stopping Criterion		**Knowing the number of clusters**	
Dataset	Mean NMI	Runtime	Mean NMI	Runtime
Yeast	0.45 ± 0.01	100 %	**0.48** ± 0.00	68.8 %
Fish	0.55 ± 0.01	100 %	**0.57** ± 0.00	49.5 %
User Know.	0.46 ± 0.01	100 %	**0.48** ± 0.00	53.0 %
Crowdsourced.	0.49 ± 0.02	100 %	**0.55** ± 0.00	52.4 %
Glass Id.	0.47 ± 0.02	100 %	**0.53** ± 0.00	50.5 %
Thyroid	0.52 ± 0.02	100 %	**0.58** ± 0.04	37.6 %
Libras Move	0.64 ± 0.02	100 %	**0.69** ± 0.00	34.8 %
Arrhythmia	0.56 ± 0.05	100 %	**0.65** ± 0.02	24.1 %

Evaluation

Results on Real World Data

Data set	Yeast	Fish	User Know.	Crowdsourced.	Glass Id.	Thyroid	Libras Move.	Arrhythmia
# of dimensions	8	463	5	28	9	13	90	278
# of classes	10	7	4	2	6	6	15	11
RandomLink max	0.48	0.57	0.47	0.55	0.50	0.54	0.68	0.61
RandomLink	**0.45** ± 0.01	**0.55** ± 0.01	**0.46** ± 0.01	**0.49** ± 0.02	**0.47** ± 0.02	**0.52** ± 0.02	**0.64** ± 0.02	**0.56** ± 0.05
SingleLink	0.12	0.03	0.05	0.03	0.07	0.51	0.12	0.35
CompleteLink	0.23	0.19	0.29	0.34	0.38	0.47	0.54	0.43
AverageLink	0.11	0.13	0.32	0.40	0.11	0.51	0.60	0.40
Ward's method	0.27	0.35	0.28	0.43	0.40	0.44	0.62	0.47
k-Means	0.27	0.28	0.23	0.43	0.43	0.46	0.59	0.44
DBSCAN	0.12	0.39	0.11	0.00	0.46	0.00	0.59	0.00
EM	0.17	0.25	0.42	0.42	0.34	0.44	0.59	0.43
Chameleon	0.00	0.47	0.35	—	0.00	0.44	0.00	0.54
Spectral Clustering	0.28	0.39	0.23	0.43	0.31	0.44	0.62	0.46
STSC	0.06	0.10	0.04	0.12	0.09	0.11	0.22	0.44
FUSE	—	0.19	0.02	0.01	0.28	—	0.18	0.31

Using Random Effects

SingleLink-Effect

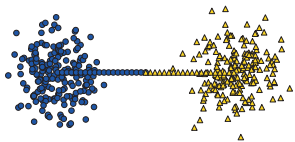

Figure: Groundtruth

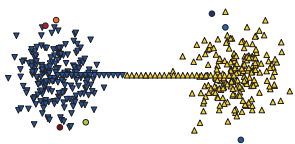

Figure: RandomLink NMI: 0.86

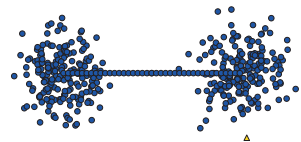

Figure: SingleLink NMI: 0.015

SingleLink-Effect

Two Gaussian cluster with a bridge in between. RandomLink separates the Link, while SingleLink cannot.

Using Random Effects

Varying Densities

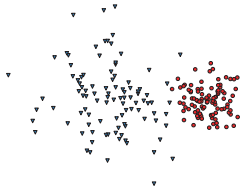

Figure: Groundtruth

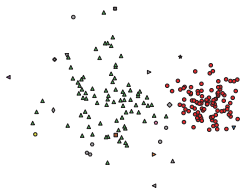

Figure: RandomLink NMI: 0.71

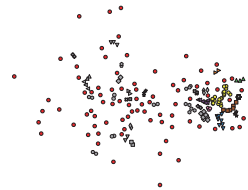

Figure: DBSCAN NMI: 0.35

Varying Densities

Two Gaussian cluster with different density. RandomLink can separate them better than the carefully parametrized DBSCAN.

Discussion

What are the findings?

RandomLink Clustering does work

- on average better than any Single Link Clustering, DBSCAN or *k*-Means
- results are stable with small deviations
- needs no parameter
- it can cluster differently shaped clusters
- proposed stopping criterion yields clustering which is close to the optimum
- doesn’t work for overlapping clusters
- not as fast as other competitors ⟶ evaluated only on ”small” datasets
- other algorithms are better if the data fits their assumption

Conclusion

Key Points and Outlook

- Did outperform classical and graph-based algorithms in the tested setting
- Performs ”equally” well as related graph-based algorithms using min-cuts
- proved to work as intended being applicable to a wide range of datasets
- including randomness can improve clustering
- more use cases, swap edges, random walks
- more theoretical work especially for the stopping criterion needed

Thank you for your attention!

Contact:

gert.sluiter@univie.ac.at
benjamin.schelling@univie.ac.at
claudia.plant@univie.ac.at

Code:

Github Repository (github.com/53RT/RandomLink)
Zip Archive

Bibliography

[1] J. Elson, J. R. Douceur, J. Howell, and J. Saul, "Asirra: a captcha that exploits interest-aligned manual image categorization.," in *ACM Conference on Computer and Communications Security*, vol. 7, pp. 366–374, 2007.

[2] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, (Sydney, Australia), 2013.

[3] R. Sibson, "Slink: An optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, pp. 30–34, 01 1973.

[4] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[5] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, pp. 364–366, 01 1977.

[6] R. R. Sokal, "A statistical method for evaluating systematic relationships.," *Univ. Kansas, Sci. Bull.*, vol. 38, pp. 1409–1438, 1958.

[7] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, pp. 49–60, 1999.

[8] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Information processing letters*, vol. 76, no. 4-6, pp. 175–181, 2000.

[9] D. R. Karger, "Minimum cuts in near-linear time," *Journal of the ACM (JACM)*, vol. 47, no. 1, pp. 46–76, 2000.

[10] G. Karypis, E. Han, and V. Kumar, "Chameleon: A hierarchical clustering algorithm of spatial data," in *Proceedings of the 8th Symposium Spatial Data Handling*, pp. 45–55, 1998.

[11] G. Karypis and V. Kumar, "A hypergraph partitioning package, department of computer science and engineering," *University of Minnesota, Minneapolis, MN*, 1998.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*.
Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.