BC-UCL - AI4E wrap-up - SP8

Document Conventions

- Font colours: Tomas black; Yaw Hon blue, Ellie Dickens green, Mohanad purple, Nathan orange. [feel free to change colours as you see fit]
- Reminders: minimalism, simplicity, directness, usefulness, helpfulness.
- Key words: UCL, UL, AI4E, wrap-up; SP8; Tomas preparations; part 1 of SP8.

Abbreviations

BC	Batch clustering.
BC-UCL	Batch clustering for UCL
CA	Clustering Accuracy
CNN	Convolutional Neural Network
DC	Design Choice
LQ	Literature Question
OS	Observation Segmentation
UCL	Unsupervised Continual Learning
UL	Unsupervised Learning

Do - Initial

- Tomas tasks:
  - (100%) Initial brainstorm.
  - (100%) Complete experimental design. 2 experiments.
  - (100%) Complete “Tomas SP8 preparations doc” with all the main heading (literature, gaps, etc.)
  - (100%) The rest of the tasks have been transferred here.
- (100%) Steps:
  - Select 1-3 folders of raw images
  - Label images in terms of observation segmentation
  - Decide on UL batch size
  - Implement UL for whole data.
  - Implement ….

Do - Tomas

- (100%) Initial basic framework and setup.
- (0%) Compile a list of pure ANN solutions to clustering. Example query: GSS [intitle:clustering intitle:neural] → 3140 results. Example paper: Du, K.L., 2010. Clustering: A neural network approach. Neural networks, 23(1), pp.89-107. Interestingly the query GSS [intitle:clustering (intitle:"neural network" OR intitle:"neural networks" OR intitle:"deep learning")] also yields 3140 results.

Think

- **Th1**. Can two or more images be efficiently fused with respect to a distance measure that is computed via a latent representation from a pre-trained neural network? This is similar to averaging two or more images with respect to the Euclidean distance, and this could be used for the merging of centroids, or even for the computation of centroids. Has this been done before? Would this involve a small gradient-based optimization process? For example, in the case of fusing two images (e.g. F = fusion(I1, I2)) relative to a pre-trained network net1, the process might involve minimising the distance(net1, F, I1) AND minimising the distance(net1, F, I2), and the fusion function might involve an element-wise weighted combination of I1 with I2, such that these weights are differentiable based on the distance losses defined above (which in turn depend on net1).
- **Th2**. Think of labels as features (not as outputs). These labels can help in the clustering process, and can influence the representation learning process. Moreover, in the context of biodiversity, labels can be hierarchical (i.e. corresponding to different taxa levels). If we think of labels this way, that is, as part of the data instance itself (i.e. as a feature), then we need to recognize the fact that most of the time this information is not available, and therefore the clustering process needs to be robust relative to missing values. Search for literature on clustering processes that explicitly tackle missing values. This idea is inspired by thinking about the human clustering process and how it often uses label information, presumably as part of the data instance itself.
- **Th3**. Consider the idea of using simple feature selection on top of the pre-trained network, in order to implement “different perspectives of what is being clustered”. Instead of doing discrete feature selection, we can use a more general feature weighting approach, where features are element-wise multiplied by a vector of weights. How can we make the clustering process differentiable, such that we can compute the error gradient with respect to these feature weights? Alternatively consider different optimization methods for searching through this weight space (e.g. genetic algorithms, particle swarm optimization, differential evolution, stochastic hill climbing, etc.). How can this search process be made more efficient? How can the evaluation function (i.e. clustering performance) be made more efficient?

Literature

Search Methods

Unsystematic but guided by questions

[This search of the literature is unsystematic and focused by specific questions. Typically, once a question has been answered, and one or more relevant references have been recorded, the search is discontinued.]

- LQ1 - 0% - What is the history of pure neural network approaches to clustering?
- LQ2 - 0% - What is the SOTA in terms of pure neural network approaches to clustering?
- LQ3 - 0% - What are the pros and cons of pure-neural vs non-neural vs neural hybrid approaches to clustering?
- LQ4 - 0% - Which pure neural network clustering approaches combine neural components for novelty estimation, similarity estimation, and other components?
- LQ5 - 0% - Which pure neural network clustering approaches explicitly consider “spectral clustering", or “agglomerative clustering”, or DBSCAN, or OPTICS, as depicted here?
- LQ6 - 0% - What makes the approaches above (LQ5) capable of clustering the examples here, when k-means (and others) fail?
- LQ7 - 0% - Has traditional “batch clustering” been explicitly adapted to UCL before? How many UCL approaches implicitly use a form of batch clustering?
- LQ8 - 0% - Do any of the “convolutional clustering” papers use a multi-scale or hierarchical approach (e.g. multiple parallel paths adopting 2D images with different resolutions)?
- LQ9 - 0% - Are there any systematic studies comparing different ways of representing a set of data points for neural clustering? Examples of ways: (1) projection into 2D spaces, (2) projection into 3D spaces, (3) tensors representing inter-point metrics (e.g. similarity matrix), (4) simple n x d matrix where n refers to the number of data points, and d refers to their dimensionality, (5) etc,
- LQ10 - 0% - Are there any Psychology-related studies on human clustering capabilities?
- LQ11 - 0% - Are there any Psychology-related studies on human clustering abilities, specifically when a subject is presented with a set of items to cluster, without being told what aspects to use for clustering? How do subjects determine the relevant aspects? This is related to feature selection. These studies could provide some interesting ideas for how to do feature selection for clustering.
- Etc.

Systematic

- [DO - Add this in due course.]

Queries

- Terms: “batch clustering”, “batch k-means”, “mini-batch clustering”, “sample-based clustering”, “threshold-based clustering”, “incremental clustering”.
- GSS ["batch clustering"] → 941 results
- GSS [“intitle:"incremental clustering"] → 675 results
- GSS ["threshold-based clustering" (intitle:review OR intitle:survey OR intitle:overview)] → 28 results
- GSS [intitle:"incremental k-means"] → 28 results
- (100%) GSS [intitle:"batch k-means"] → 26 results
- GSS [intitle:"adaptive k-means" "number of clusters" ("new cluster" OR "new clusters")] → 25 results
- GSS [“batch clustering" (intitle:review OR intitle:survey OR intitle:overview)] → 22 results
- (100%) GSS [intitle:"batch clustering"] → 14 results
- GSS ["batch clustering" "continual learning"] → 2 results

Seed Papers

- Liao, K., Liu, G., Xiao, L. and Liu, C., 2013. A sample-based hierarchical adaptive K-means clustering method for large-scale video retrieval. Knowledge-Based Systems, 49, pp.123-133.
  - Assuming that sample-based clustering is based on random samples, then this is different from our sequential (online; throw away) mini-batch approach. It is of course very closely related, but it is different.
  - [DO - 0% - Have a look at all of the in-citations.]
- Ismkhan, H., 2018. Ik-means−+: An iterative clustering algorithm based on an enhanced version of the k-means. Pattern Recognition, 79, pp.402-413.
- Newling, J. and Fleuret, F., 2016. Nested mini-batch k-means. Advances in neural information processing systems, 29.
- Béjar Alonso, J., 2013. K-means vs mini batch k-means: A comparison.
- Alguliyev, R., Aliguliyev, R., Bagirov, A. and Karimov, R., 2016, October. Batch clustering algorithm for big data sets. In 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-4). IEEE.
  - Differences: (1) we need adaptive cluster numbers, (2) (check) how information is used (i.e. centroids) seems to be different in this seed paper.
- Alguliyev, R.M., Aliguliyev, R.M. and Sukhostat, L.V., 2021. Parallel batch k-means for Big data clustering. Computers & Industrial Engineering, 152, p.107023.
  - Differences: this paper involves parallel batching (without dependencies (e.g. information sharing (e.g. centroids))) between batches.

Other Papers

- Gallé, M. and Renders, J.M., 2012, April. Full and mini-batch clustering of news articles with star-em. In European Conference on Information Retrieval (pp. 494-498). Springer, Berlin, Heidelberg.
- Yamini, G. and Devi, B.R., 2018. A new hybrid clustering technique based on mini-batch K-means and K-means++ for analysing big data. Int. J. Recent Res. Aspects, pp.203-208.
- Kushwaha, A.P.S., Jaloreeb, S. and Thakurc, R.S., 2021. A comparative review of incremental clustering methods for large dataset. International Journal, 10(2).
- Possible examples of bio-inspired papers:
  - https://arxiv.org/abs/1706.05048

Gaps

- No UCL approaches applied to biodiversity auto-curation yet.
- [Check] No UL approaches applied to biodiversity auto-curation yet.
- [Check] No studies of UL or UCL used for OS and CA simultaneously.
- [Check] No systematic and explicit studies on batch clustering applied to UCL. Note that many-[CH] UCL approaches use some form of batching, so this gap needs to be worded very carefully.
- [Check] Are there any studies about “batch clustering” with adaptive numbers of clusters?

Random Related Raw Ideas

- **RRRI1**. The following idea could potentially be called: multi-projection convolutional clustering. The idea is to take multiple 2D projections of high-dimensional data, cluster each of these projections using a “clustering CNN”, and then combine these clusterings somehow in order to generate that final clustering corresponding to the original high-dimensional data. One of the key points here would be on how best to select the different projections. For instance assume that each projection is built from selecting two nodes of a pre-trained representation: how should these pairs of nodes/features be selected? Moreover, the combination of information from the different projections can be done at different levels, e.g.: (1) at the level of clustering outputs (i.e. the actual 2D clusters) (as mentioned above), (2) at the level of creating new 2D representations based on the relationship of different 2D projections (e.g. this could be related to: how do the inter-point distances change with different projections).

Preliminary Framework Setting

- No more work is needed here. I leave this section here, for the sake of reference.

Research questions

- RQ1: Is UL/UCL useful for auto-curation of biodiversity images?
- RQ2: Is a batch clustering approach viable as a UCL approach?
- RQ3: Can batch clustering be improved by one or more simple glue measures, in order to improve OS and CA performance in the UCL setting?

Experimental Design

- As of 12/05/22, the experiments below are obsolete, since I decided to use a simpler overall design. Each student project (for the summer of 2022) should focus on one or more design choices (DCs); see the section below.

Threshold-based k-means

- Use the same simple thresholding approach reported in (Bhatia, 2004):
    - Step 1. “Select an element from the given data set. This element is assigned as the seed of a cluster by itself.”
    - Step 2. “For every unclassified element, find its distance from the centroid of the existing clusters. If the distance is less than the threshold, assign the element to this cluster. Recompute the centroid of the cluster as the average of all properties of all elements in the cluster. If no such cluster can be found after examining all current clusters, assign the element as the seed for a new cluster.”
    - Step 3. “If, as a result of the above step, the distance of new cluster to another cluster is smaller than the threshold, merge the two close clusters together, and recompute the cluster distances.”
    - Step 4. “The algorithm stops after all the elements have been assigned to one or the other cluster.”
- Consider tweaking step 3. Is it needed in this context? Might it destroy specimens from previous batches?

Experiment 1 (addresses RQ1-RQ2):

- Preparation:
    - Two datasets, each one with 3 folders, one for tuning the threshold, the other for testing.
- Conditions:
    - Preparation. Tune threshold based on “threshold tuning data set”. Use this same threshold in all conditions.
    - Condition 1. Basic threshold-based UL applied to whole data (3 folders). No time-stamps.
    - Condition 2. UL applied to batches with batch i clusters initialized with batch i-1 clusters. Images from batch i-1 do not participate in the clustering of images in batch i, however, all images are kept for final performance metrics. No time stamps.

- Performance metrics e.g.: observation segmentation (OS), clustering accuracy (CA) (pseudo-labels to real-labels), processing time, and other general cluster metrics (e.g. purity, etc.).
- Hypotheses:
    - Condition 1 is generally more accurate (i.e. OS and CA) than condition 2.
    - Condition 2 performance is well above chance level, and is reasonably similar to condition 1, therefore a threshold and batch-based k-means approach is a viable solution for UCL in the domain of auto-curation of biodiversity images.

Experiment 2 (addresses RQ3):

- General notes:
    - Experiment with different modifications to the glue between batches in order to see how these modifications affect OS and CA. One key constraint: the modifications should not modify the UL/clustering itself. They should only work with the outputs and inputs (e.g. centroids).
    - All conditions in this experiment are based on batch clustering.
- Example modifications (a subset of these can be chosen):
    - M1. A random sample of images from batch i-1 are used in the clustering of batch i.
    - M2. Similar to M1 except that the sample is not random, but rather guided by some criterion (e.g. maximize diversity of instances).
    - [Add more modifications here.]
- Conditions:
    - Condition 1: no modifications. Centroids from batch i-1 are used to initialize centroids of batch i.
    - Conditions 2-n: modifications.

Follow-up experiments / Future Work

- [DO - 0% - as of 21/04/22 - check the list below, since some points have been experimented with already.]
- How many clusters can the approach deal with before running into memory issues, given specific memory constraints?
- Add image timestamps for clustering.
    - RQx: What is the impact of including timestamps in the clustering process?
- Compare different distance metrics.
- Improve clustering quality measures. Improve selection of “best k”, in terms of accuracy and speed.
- Use pre-trained networks for feature extraction, for distance measures. For example, compare: (1) network trained on ImageNet, (2) networks trained on ImageNet but fine-tuned for iNaturalist, (3) networks trained on iNaturalist.
- Incorporate ANN distance metrics into (1) centroid re-computation, and (2) centroid fusion.
- Try to automate threshold computation.

- Experiment with other forms of inter-batch glue. What other types of information should one pass from batch to batch (e.g. centroids, variances, patterns, etc.).
- All the standard weaknesses of k-means ar inherited here, e.g.: no cluster overlap, spherical clusters, clusters of the same size, etc. Address these weaknesses.
- Experiment with different "clustering quality" metrics for finding k.

Design Choices - UCL framework

Initial Notes

- This section should be the main focus for all/most UCL-related student projects for the summer of 2022.
- The main sub-folder with the target code is: T-K-Means-UCL-AutoCurat.
- As of 08/05/22 the preliminary experimental framework for batch clustering (targeting UCL) has been set up, which means that all efforts are based on improving OS accuracy, by working on the design choices below (or additional ones).
- Different design choices (DC) for the simple and general UCL framework:
    - [Note: do not change the labels (e.g. "DC-1") since they exist in external references. New labels can be added, but don't change the existing ones.]
    - DC1. Existence or not of specimen detection (with cropping around the specimen). Specimen detection algorithm or model. As of 30/04/22 I am not using any specimen detection (the whole image is used).
    - DC2. Different clustering algorithms (e.g. k-means vs. spectral clustering).
    - DC3. Different cluster initializations.
    - DC4. Different distance metrics to be used within the chosen clustering algorithm.
    - DC5. Different clustering quality estimators (i.e. how to select the best k).
    - DC6. Different search procedures for optimal k (e.g. grid search is a baseline).
    - DC7. Different pre-trained networks for feature extraction (e.g. ImageNet vs. iNaturalist).
    - DC8. Different feature extraction approaches (e.g. one or more layers; which layers; etc.).
    - DC9. Different batch-clustering/UCL logic.
- Examples of "different batch-clustering/UCL logic":
    - (0) perform clustering in batches separately and then (at the end) find out which centroids are similar enough to each other to be considered the same species. Here there is no centroid fusion; just a list keeping track of which centroids are similar enough to each other.
    - (1) perform clustering in batches separately and then fuse centroids that are similar, i.e.: (i) cluster batch b, (ii) cluster batch b+1, (iii) fuse similar centroids, (iv) continue.
    - (2) link batches; use frozen centroids from batch t-1 in batch t.

Brainstorming Design Choice Variations

DC1. Existence or not of specimen detection

- Example variations:
    - No specimen detection and cropping (around the specimen). Here the whole image is used.
    - Specimen detection:
        - Plugged in MegaDetector.
        - Fine-tuned YOLO detectors.
        - Other state of the art deep learning object detection architectures.
        - New architecture.

DC2. Different clustering algorithms

- Many variations are possible:
    - Variants of k-means.
    - Some candidates from scikit-learn:
        - Spectral clustering.
        - Agglomerative clustering.
        - DBSCAN.
        - Optics.
        - Etc.
    - Some pure neural network solutions:
        - SOM
        - ART
        - Mixtures of autoencoders (example paper)
        - Etc.
    - Etc.

DC3. Different cluster initializations.

- Example variants:
    - Random initialization
    - Forgy initialization
    - Random partition initialization
    - K-means++ initialization
    - Etc.

DC4. Different distance metrics

- Example distance metrics:
    - Euclidean Distance.
    - Manhattan Distance.
    - Minkowski Distance.

  - Chebyshev Distance
  - Cosine Similarity
  - Jaccard Similarity
  - Etc.

DC5. Different clustering quality estimators

- Example metrics:
  - Davies-Bouldin Score
  - Calinski Harabasz Score
  - Dunn Index
  - Silhouette Coefficient
  - Etc.

DC6. Different search procedures for optimal k

- Examples:
  - Simple grid search.
  - Hierarchical grid search.
  - Random search.
  - Hill-climbing or other simple heuristic approaches.
  - Global stochastic optimization algorithms.
  - Neural network to estimate the optimal k (input: whole dataset projected into n low dimensional spaces (like the 2D spaces in section 3.2.1 here); output: mapping from low-dimensional spaces to k). For an intuitive notion, think of the 2D scatterplots in section 3.2.1 here, and think of a CNN that takes as input the image of a scatterplot and returns the number of clusters for that scatterplot.
    - Check whether this has been done before (check the literature). This could be very impactful. It is easy to come up with self-supervised data for training such networks. The architecture might be more challenging to get right. Example paper: https://arxiv.org/pdf/1706.05048.pdf. Search for other papers.
  - Etc.

DC7. Different pre-trained networks for feature extraction

- Examples (used as-is or fine-tuned):
  - Different networks pre-trained on ImageNet.
  - Different networks pre-trained on iNaturalist.
  - Different networks pre-trained on other relevant datasets.
  - Different aspects of pre-training and fine-tuning. Different transfer learning ideas.
  - General advice: experiment with other pre-trained networks; fine-tune networks; etc; in particular move towards models that have been exposed to iNaturalist data.

DC8. Different feature extraction approaches

- Examples:
  - Final hidden FC layer.
  - Some other FC layer.
  - Some combination of FC layers.
  - Some combination of FC and convolutional layers.
  - Some combination of FC and convolutional layers, and some function of layers.
  - Feature selection applied to one or more extracted layers.
    - This is a very simple idea which should give a big boost to the performance. Some features are relevant to our targets whereas others are not; the latter should make clustering harder; clustering is always relative to the task and/or features of interest (this is the same when humans perform clustering too, e.g. “From what perspective do you want me to cluster these items?”). [DO - 0% - Check the literature for feature selection for clustering (in DL and outside DL). The conservative query [intitle:"feature selection" intitle:clustering] yields 1090 results. This is a highly-populated area.]
      - More conservative query → GSS [intitle:"feature selection" intitle:clustering ("deep learning" OR "neural networks")] since 2020 → 101 results.
  - Etc.

DC9. Different batch-clustering/UCL logic

- Examples:
  - No information shared between batches. No global/final analysis of centroids (e.g. fusion of sufficiently similar centroids).
  - No information shared between batches. Global/final analysis of centroids (e.g. fusion of sufficiently similar centroids).
  - Different variants of information shared between batches, e.g.:
    - Batch b+1 centroids are initialized and expanded with batch b centroids.
    - Etc.

Sub-projects (SP) (i.e. papers, experimental designs)

Notes

- All sub-projects must include at least 50% of systematic experimentation, at a depth and level of detail that could be classified as publishable.
- Sub-project assignments:
  - Ellie → to be confirmed.
  - Yaw Hon → to be confirmed.
  - Mohanad → SP2 focusing on DC7 (pre-trained networks).

List of SPs

- **SP1 (1 sub-project)**. Experiment with BC-UCL framework freely (any DC or combination of DCs) attempting to improve OS and clustering accuracies. This project has two phases: (1) (phase 1) fully exploratory, (2) (phase 2) systematic experiments based on phase 1.
- **SP2 (roughly 10 sub-projects)**. Commit to, and experiment with, one or two specific DCs with the aim of improving OS and clustering accuracies. Examples: SP2-DC1, SP2-DC2, etc.
- **SP3 (1 sub-project)**. Work on a specific sequence of DCs, e.g.:
  - Start with the basic BC-UCL framework.
  - Try to improve OS and clustering accuracy by systematically investigating different solutions for DC7.
  - Once the best solution has been obtained by optimising DC7, then systematically optimise in sequence: DC8, DC4, DC2, DC3, DC9, DC6, DC5, and DC1.
- **SP4 (more than 10 sub-projects)**. Work on a specific aspect of a specific design choice, e.g.: work on the feature selection or feature weighting solution of DC8.

References

- Bhatia, S.K., 2004, May. Adaptive K-Means Clustering. In FLAIRS conference (pp. 695-699).