BCubed Revisited: Elements Like Me

van Heusden, R.; Kamps, J.; Marx, M.

DOI
10.1145/3539813.3545121

Publication date
2022

Document Version
Final published version

Published in
ICTIR'22

BCubed Revisited: Elements Like Me

Ruben van Heusden
r.j.vanheusden@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

Jaap Kamps
kamps@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

Maarten Marx
maartenmarx@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

BCubed is a mathematically clean, elegant and intuitively well behaved external performance metric for clustering tasks. BCubed compares a predicted clustering to a known ground truth through elementwise precision and recall scores. For each element, the predicted and ground truth clusters containing the element are compared, and the mean over all elements is taken. We argue that BCubed overestimates performance, for the intuitive reason that the clustering gets credit for putting an element in its own cluster. This is repaired, and we investigate the repaired version, called "Elements Like Me (ELM)". We extensively evaluate ELM and conclude that it retains all positive properties of BCubed and gives a minimum 0 zero score when it should.

CCS CONCEPTS

• **Information systems → Evaluation of retrieval results**.

KEYWORDS

Information Retrieval, BCubed, Clustering, Metrics

ACM Reference Format:
Ruben van Heusden, Jaap Kamps, and Maarten Marx. 2022. BCubed Revisited: Elements Like Me. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '22), July 11–12, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3539813.3545121

1 INTRODUCTION

We review the external clustering performance metric *BCubed* [1, 2], show a flaw and propose a repair. We then evaluate the repair theoretically and experimentally and show that the proposed repair yields more intuitive results, particularly for the F1 score.

To keep this paper short, we refrain from reviewing clustering methods and clustering evaluation measures. For these, we refer to [8] and [1], respectively.

In essence clustering and (single label) classification perform the same task: given a set of items D, they partition D. But when it comes to evaluation with comparison to a gold standard, things are very different.

With classification, the number of blocks in the partition is known (the set of labels), and a mapping exists between the true blocks and the predicted blocks (namely the identity mapping on the labels). So, counting errors is straightforward by making the cross table of predicted and gold truth values (the *confusion table*), and computing precision and recall as the diagonal divided by the two margins, respectively.

With clustering, there is (at prediction time) no known number of blocks (as the label set is unknown), and there is no mapping between the predicted blocks and the true labels. This makes counting errors much less straightforward, witnessed by the numerous proposals on how to do this, nicely surveyed and classified by Amigó et. al. [1].

1.1 BCubed P, R and F1

The most natural and easy to use and understand measure is the BCubed measure, proposed by Bagga and Baldwin [2]. Besides that, Amigó et. al. [1] define four intuitive desiderata for a clustering performance metric and BCubed is the only one satisfying all of them.

BCubed is defined as follows. Let D be a set of items and $f : D \longrightarrow \mathcal{P}(D)$ a partition with the property that $e \in f(e)$ for all $e \in D$. Thus f assigns to each element e in D a subset of D containing e, subsets do not overlap, and together cover D. Note that f can be seen as a clustering, or as a labelling of elements in D with the sets in $range(f)$.

Assume we have two partitions: the predicted f_p, and the gold standard truth f_t. For each $e \in D$, define the precision, recall and harmonic mean F1 of e as usual:

$$P(e) = \frac{|f_p(e) \cap f_t(e)|}{|f_p(e)|} \tag{1}$$

$$R(e) = \frac{|f_p(e) \cap f_t(e)|}{|f_t(e)|} \tag{2}$$

The harmonic mean of P and R is usually defined as $2PR/(P+R)$, but we use here the equivalent direct definition in terms of the quadrants in the confusion table.

Let $TP = |f_p(e) \cap f_t(e)|$, $FP = |f_p(e) \setminus f_t(e)|$ and $FN = |f_t(e) \setminus f_p(e)|$.

$$F1(e) = \frac{TP}{TP + \frac{(FP+FN)}{2}} \tag{3}$$

We can use the symmetric difference $A \ominus B$ to give an equivalent definition, from another intuitive angle, of the number of misclassifications, and obtain this definition of the F1 score per element:

$$F1(e) = \frac{|f_p(e) \cap f_t(e)|}{|f_p(e) \cap f_t(e)| + \frac{|f_p(e) \ominus f_t(e)|}{2}} \tag{4}$$

The BCubed precision, recall and F1 values of the *predicted partition* are simply the means over all elements in D.

ICTIR '22, July 11–12, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545121

2 FLAW OF BCUBED

The intuitive interpretation of the evaluation measures restricted to an element e is that they indicate how well the classifier is able to *find the elements similar to e*. The precision $P(e)$ equals 1 if it did not assign wrong elements to the block of e and the recall is 1 if it assigned all elements in the true block of e to the predicted block of e.

However, if one thinks of BCubed in this way it becomes clear that the measure is giving too much credit. $P(e)$ and $R(e)$ should measure how well the *other elements* in D are assigned to the block of e. But the definition also counts e itself, which by definition is assigned to the (correct) block of e.

This flaw is exemplified by the observation that because e is always an element of $f_p(e)$ and $f_t(e)$, the numerator $|f_p(e) \cap f_t(e)|$ is never equal to 0, and thus none of $P(e), R(e)$ and $F1(e)$ can ever be equal to zero.

Now consider this simple concrete example: $D = \{1, 2\}$ and f_t assigns each element to D. Now let f_p partition D into $\{1\}$ and $\{2\}$. This is the only other partition that can be made, and it is wrong. However, for both elements e, $P(e) = 1$, as it makes no mistakes, $R(e) = .5$, as half of the true elements of the block of e are in its predicted block, and so $F1(e) = .66$. Taking the mean over all elements, we get an $F1$ value of .66 for this prediction. Having only two outcomes for the prediction, correct and wrong, we see here that BCubed F1 does not nicely separate these two predictions. The dual example on the same set D yields $P(e) = .5$, $R(e) = 1$ and again $F1(e) = .66$ for each e, and thus also for the predicted partition. It is desirable that the two *extreme clusterings* (assign nobody and assign everybody) receive the same score, but it would be better if they both received a much lower $F1$ score, preferably 0. Note that if we put the two examples together in one *testset*, and average over the scores per test-items, both extreme clusterings again receive the same scores of $P = R = .75$ and $F1 = .66$. This example can be generalized to sets D of cardinality n. Then the "singleton clustering" has $P = 1$, $R = 1/n$ and $F1 = \frac{2}{n+1}$, and the "all in one clustering" the dual $P = 1/n$, $R = 1$ and $F1 = \frac{2}{n+1}$.

We can **repair** this by not counting the element e itself, forcing the recall in the singleton clustering example to become 0, as it should be, because the classifier failed to find any similar element to e. Then $F1$ becomes 0 as well, resulting in a much better and fairer estimate of the quality of this prediction. To repair this we only need to subtract the set $\{e\}$ in both numerator and denominator in the above definitions:

$$P(e) = \frac{|f_p(e) \cap f_t(e) \setminus \{e\}|}{|f_p(e) \setminus \{e\}|} \tag{5}$$

We have to account for the case when $f_p(e) = \{e\}$, which would result in a division by zero. Naturally, we set $P(e)$ equal to 1 in this case (no others are assigned to the cluster of e, so no mistakes have been made). We do the same for recall, setting $R(e) = 1$ if $f_t(e) = \{e\}$.

$$R(e) = \frac{|f_p(e) \cap f_t(e) \setminus \{e\}|}{|f_t(e) \setminus \{e\}|} \tag{6}$$

And for $F1$ we only change the definition of the true positives TP into $|f_p(e) \cap f_t(e) \setminus \{e\}|$, and stipulate that $F1(e)$ equals 1 if $f_t(e) = f_p(e) = \{e\}$.

It is immediate that a perfect prediction still receives only ones. The wrong singleton clustering on any one block ground truth still has a precision of 1 (because it does nothing, so makes no mistakes), but now a recall and thus also F1 of 0. The dual "all in one class" prediction on the singleton gold truth has maximal recall but at the cost of zero precision and thus also zero F1. Thus this simple repair gives the two extreme "baseline" clusterings on all extreme examples exactly the same minimal $F1$ value of zero. Much nicer than the diminishing sequence for sets of increasing cardinality.

2.1 A new name

In the rest of the paper, we further evaluate this repair. But let us first give it a name. The BCubed measure was introduced by Bagga and Baldwin [2]. In a footnote they attribute the idea of BCubed to Bierman, and thus the cubed Bs. We opted for *ELM*, an abbreviation of *Elements Like Me*, which is a good mnemonic of the way we compute the repaired BCubed measure.

3 EVALUATION

We evaluate our ELM metric in four ways:

(1) We show that ELM is conservative: ELM is always less than or equal to BCubed.
(2) We compare ELM to BCubed on a number of fixed baselines on a real world clustering test set.
(3) We do the same comparison but now using hierarchical clustering.
(4) We show that ELM still satisfies the four constraints satisfied by BCubed introduced in [1].

The totality of tests show that ELM retains all good properties of BCubed, but is better in separating good from bad clustering methods.

3.1 ELM is conservative

We compare the BCubed and ELM versions of P, R and $F1$ using superscripts P_B, P_{ELM}, etc.

Claim. Let D be a set and f_t and f_p be the ground truth and predicted partition of D, respectively. Then for each $e \in D$, for each metric $O \in \{P, R, F1\}$, $O_{ELM}(e)$ is strictly smaller than $O_B(e)$, except when both are equal to 1, and that is when $f_t(e) = f_p(e)$ or the condition in the definition of ELM to avoid zero division applies.

Proof. The claim follows from the following fact:

if $m < n \neq 1$, then $\frac{m-1}{n-1} < \frac{m}{n}$ (*).

Let D, f_t and f_p be as in the claim and $e \in D$ arbitrary. We show the claim for precision. The arguments for recall and $F1$ are similar. If $f_t(e) = f_p(e)$ or $f_p(e) = \{e\}$, $P_B(e) = P_{ELM}(e) = 1$. So assume otherwise. Then $|f_p(e)| > 1$ and $|f_p(e) \cap f_t(e)| < |f_p(e)|$. In this situation the sole difference in the definitions of BCubed and ELM is that we subtract 1 from both the numerator and the denominator. The claim now follows from (*).

3.2 ELM vs BCubed on a real testset

We take a clustering dataset for which we have ground truth and compare ELM and BCubed scores for a number of predictions. We first look at simple fixed cluster size baselines and observe what

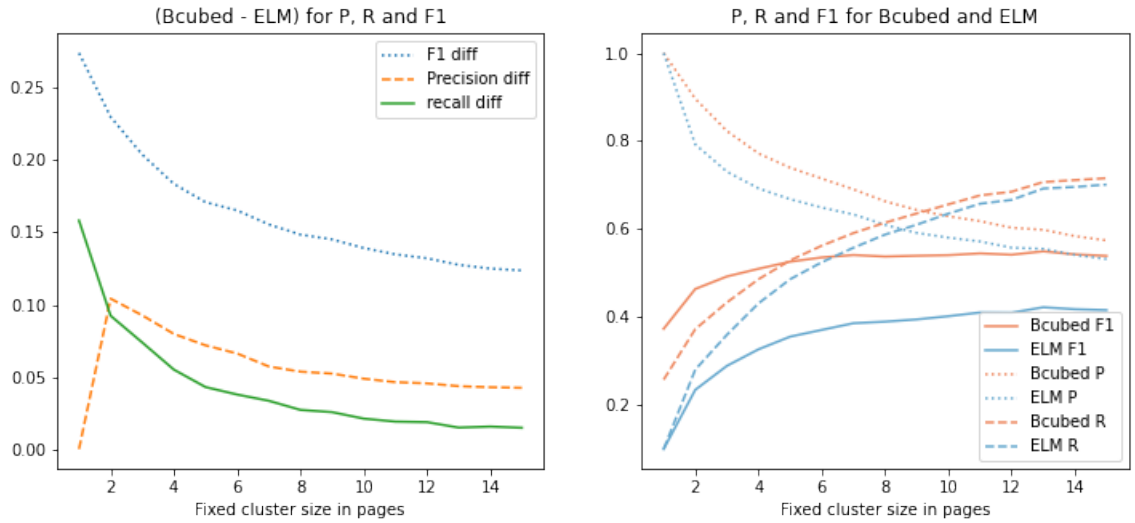

Figure 1: Difference of mean average P, R and F1 between BCubed and ELM (left) and mean average P, R and F1 scores for BCubed and ELM (right) for the experiment with clusters of a fixed number of pages (N=75).

happens when we vary the cluster size. Then we look at hierarchical clustering and compare the scores of BCubed and ELM over the whole dataset for a specific version of a hierarchical clustering algorithm

Following [1, 2], we report the mean average F1 scores. Thus for every sample D in our testset, we take the average over the $F1(e)$ for each $e \in D$, and then we take the mean over all samples in the testset.

We have a (test)set C of samples S each consisting of items which need to be clustered. This specific set has 75 samples with in total 2.508 true clusters over in total 20.618 elements. The mean and median cluster size is 33 and 17, respectively. Each sample is a sequence of pages of text divided into documents. Thus each cluster consists of a document, which is a continuous sequence of pages. The elements are thus the pages. This scenario is common in the field of Page Stream Segmentation [7, 9].

3.2.1 Fixed baselines. For the comparison of the P, R and F1 scores of BCubed and ELM for a fixed baseline, we used the dataset described above, and varied the size of the clusters from 1 to 15 pages. We also compare BCubed and ELM in the case of the two extreme clusterings: only singleton clusters and putting all elements in the same cluster.

Table 1 shows the results of evaluating the extremes. We can see that ELM gives lower scores for both of these extremes, with the difference for recall, and hence also for F1, of the 'singleton clusters' variant being large. This is because for all non singleton clusters in the gold standard, no similar elements are found if all predicted clusters are singletons because of the exclusion of the item itself. So for most elements the ELM recall will be equal to zero. In fact for this prediction, the ELM recall is equal to the proportion of single page clusters in the dataset.

Table 1: Mean average BCubed and ELM precision, recall and F1 scores for the two extreme clusterings: all elements are put into one cluster, and each singleton is a cluster (N=75).

All Singletons			
	Precision	Recall	F1
BCubed	1.0 (σ = 0.0)	0.26 (σ = 0.20)	0.34 (σ = 0.22)
ELM	1.0 (σ = 0.0)	0.10 (σ = 0.13)	0.10 (σ = 0.13)
All in one cluster			
	Precision	Recall	F1
BCubed	0.26 (σ = 0.28)	1.0 (σ = 0.0)	0.34 (σ = 0.28)
ELM	0.24 (σ = 0.27)	1.0 (σ = 0.0)	0.31 (σ = 0.28)

The difference in mean and standard deviation is much smaller for the 'one giant cluster' prediction. This also holds when we consider the distribution of all scores (Figure 2). Also notice that the distribution for the singleton clusters prediction has less variance and skew for ELM.

Figure 1 shows that the difference between BCubed and ELM is largest for the F1 score on the fixed page baseline. For all three metrics the difference decreases as the fixed number of elements in the cluster increases, but for F1 the difference remains large. This further proves the point that the ELM score is more conservative than the BCubed score.

3.2.2 Hierarchical clustering. We now expand on the comparison of fixed baselines conducted in the previous section by using a constrained (clusters must consist of consecutive pages) hierarchical clustering algorithm, employing cosine similarity between

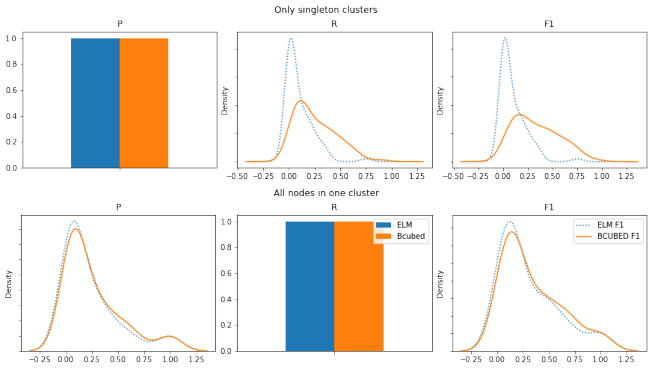

Figure 2: Kernel density estimation plots for mean average ELM and BCubed metrics computed for the two extreme clusterings (N=75).

Table 2: Mean average P, R and F1 scores for the constrained hierarchical clustering approach (N=52).

	P	R	F1
BCubed	0.69	0.69	0.51
ELM	0.65	0.66	0.44

elements. Pages are represented as character 2- and 3-grams frequency vectors[1]. The number of predicted clusters is determined by the distance between two clusters compared to the distances at the same and possible lower levels of the dendrogram. This method is known as the *inconsistency coefficient* and many variants exist for determining a threshold.

For our dataset, we found that a coefficient of 1.5 times the mean of the lower level distances worked best, with the mean taken over all distances below the cluster decision in the dendrogram. Results in Table 2 show little difference between BCubed and ELM precision and recall, but quite a large difference in their harmonic mean. The density plots in Figure 3 show that the variance in difference between the scores is also the largest for F1.

Figure 4 shows that the difference between the mean F1 score per document of BCubed and ELM decreases as the average cluster size in the gold standard increases. This is expected as the subtraction of $\{e\}$ in ELM has a larger effect when the denominator / cluster size is smaller. When we used the median cluster size instead of the mean, the plot was even more skewed towards large differences with a low median number of clusters.

3.3 ELM satisfies the constraints of Amigó et al

We show that the constraints presented by Amigó et. al. [1] hold for the ELM F1 metric. This paper shows that the family of BCubed like cluster evaluation metrics is the only one satisfying all their 4 constraints. For a thorough explanation and motivation of the four constraints we refer to the original paper. We follow the same line of reasoning as in [1] and also use their informative pictures.

[1]For this experiment, we needed the text of the documents, which was only available for 52 of the 75 streams.

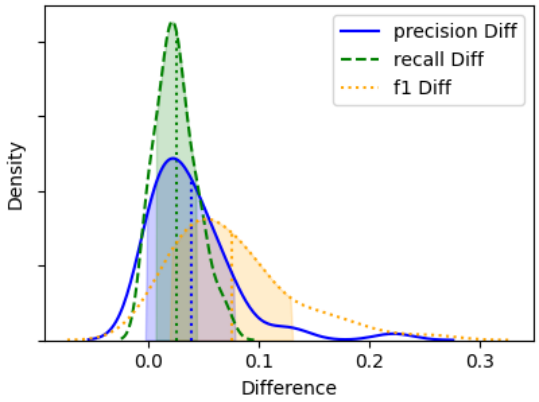

Figure 3: Density plots for difference between BCubed and ELM (BCubed - ELM) for average P, R and F1 for the hierarchical clustering experiment (N=52). The dotted lines represent the means of the respective difference and the colored areas the standard deviations.

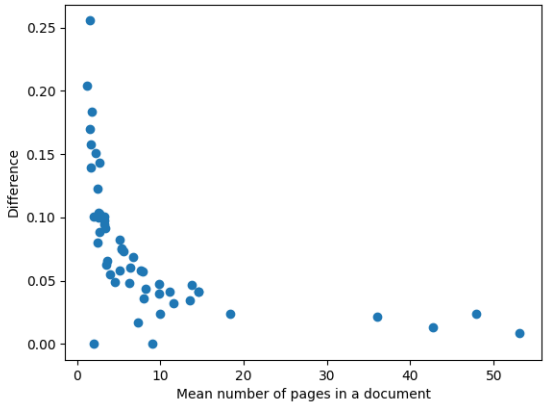

Figure 4: Scatter plot showing BCubed F1 - ELM F1 plotted against the mean number of pages in a document (N=52).

3.3.1 Homogeneity. The homogeneity constraint states that a cluster assignment D_1 that splits samples into homogeneous subgroups should be scored higher than an assignment D_2 that mixes samples of different subgroups together, like in Figure 5 in Appendix A.1.

The ELM recall for each element is the same in D_1 and D_2, but the precision is lower for the elements in the mixed cluster in D_2, than in the homogeneous clusters in D_1. Hence, the mean ELM $F1$ score of D_1 is higher.

3.3.2 Completeness. The cluster completeness states that a cluster assignment D_1 that groups items belonging to the same cluster together should receive a higher score than a clustering D_2 that subdivides items from a homogeneous cluster, like in Figure 7 in Appendix A.3.

The argument is the dual of the previous. Here precision is maximal for all elements in both partitions as all clusters are homogeneous. But ELM recall is lowered for those elements in the separate C_2 and C_3. In fact, recall is with ELM even 0 for singleton clusters. Thus the mean ELM $F1$ is higher for the partition D_1 with the joined clusters.

3.3.3 Rag Bag. The Rag Bag constraint states that adding a singleton cluster to a cluster consisting of all differently labeled elements, a *rag-bag*, should score better than an assignment adding this singleton to a homogeneous cluster, as in Figure 6 in Appendix A.2. In this example, this means that D_1 should score better than D_2.

First observe that all elements have the same recall in both clusterings. Now the element in C_3 has the same precision of 0 when it is added to C_1 or to C_2. The elements in the rag-bag C_2 also keep the same precision (namely 0) irrespective whether C_3 is joined or not. But those in the homogeneous C_1 see a drop in precision (from 1 to $\frac{3}{4}$) when C_3 is joined. Thus D_1 has a higher mean $F1$.

3.3.4 Cluster Size vs. Quantity. The clustering size vs. quantity constraint states that making a small error in a big cluster should be favorable to making many small mistakes in small(er) clusters. To show it, consider a set D containing a subset of 4 special elements. In case 1, all these 4 have the same label and these are the only 4 with that label, and can be either all in one cluster or divided into a 3 size cluster and a singleton. If they are all together, all 4 elements have recall of 1. If they are divided, the singleton has recall 0, and the other 3 all recall of $\frac{2}{3}$. The precision of all 4 equals 1. Thus when the 4 elements are in one cluster, the F1 of each equals 1. But when divided, the singleton has F1 score of 0, and the other three $(2 \cdot 1 \cdot \frac{2}{3})/(1 + \frac{2}{3}) = 4/5$, so the sum of F1 scores drops with $4 - \frac{12}{5}$.

On the other hand, consider that these 4 elements are labeled 2 by 2 with 2 labels, and these 4 are the only elements with these labels. Then also if the clustering is correct all 4 elements have recall of 1. If we split both clusters, all elements have recall of 0. Again in both cases all elements have a precision of 1. So here the the numerator in the mean F1 score drops with $4 - 0 = 4$. And thus the mean F1 score of the first case, a small error in a big cluster, is larger than the score in the second case, two errors in two small clusters.

4 OTHER REFINEMENTS OF BCUBED

Since the introduction of BCubed, several refinements have been proposed to adapt the metric for specific use cases. Moreno and Dias [6] proposed two adjustments to the BCubed F1 metric that is more suited for usage with highly unbalanced datasets, such as image clustering of ambiguous search terms on the web. They argue that the standard version of BCubed is less suited for this, because the larger clusters (of the non interesting class) have an unreasonable effect on the total score, comparable to the unreasonableness of accuracy in such cases. Both proposed alterations have the effect of weighting precision more than recall. The most straightforward one is not to use the harmonic mean F1, but a differently weighted average.

In addition to proposing the adjustment to the BCubed metric, the authors also show that it satisfies the *unbalanced* constraint from [4]. This constraint states that cluster assignments misplacing elements from small clusters into large clusters should be penalized more than cluster assignments misplacing elements from a large cluster into smaller clusters. In their work, they also propose an adjustment to the Rand index that satisfies this constraint.

The original BCubed metric is not well suited to cases where elements can simultaneously belong in multiple clusters, for example clustering news articles into different categories, where a news article might be associated with multiple topics or tags. An extension to BCubed that handles the case of overlapping clusters is proposed in [1], but this extension might assign the maximum F1 score to a clustering that is not exactly equal to the gold standard. The ELM metric suffers from the same problem as BCubed in the case of overlapping clusters, because in this case the numerator and denominator in BCubed are always equal, leading to a perfect score. Obviously in this situation, subtracting 1 from both numerator and denominator still yields a perfect score. Morena and Dias [6] propose *CICE-BCubed*, which fixes the aforementioned issue for BCubed by also checking for pair occurrences in different classes.

5 CONCLUSION

We indicated that the BCubed F1 measure gives an overestimation of the performance of a clustering method, repaired the definition, and evaluated the result positively. The new ELM measure retains all the positive properties of BCubed, is a better performance indicator, and is better able to separate different approaches tested on the same testbed.

We end with looking at the problem from the perspective of network science [3, 5]. If we view a clustering not as a set of subsets on some domain D but as a *binary relation on* D, we take a network perspective. A clustering or partition then corresponds to an equivalence relation E. The neighbor function $f(e) = \{e' \in D \mid eEe'\}$ then is the clustering function used to define BCubed. In network science, it is customary to work with simple (that is, irreflexive), and if possible, undirected relations. If we replace the equivalence relation with this irreflexive undirected relation, we end up with the same partition (in network science the blocks are called *cliques*). But on this network, the same neighbor function defines ELM, simply because no element is a neighbor of itself. We can speculate how BCubed would have been defined if one of the three B's had been a network scientist.

ACKNOWLEDGMENTS

This research was supported in part by the Netherlands Organization for Scientific Research through the ACCESS project grant CISC.CC.016 (https://www.nwo.nl/en/projects/ciscc016).

REFERENCES

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval* 12, 4 (2009), 461–486.

[2] Amit Bagga and Breck Baldwin. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1* (Montreal, Quebec, Canada) *(ACL '98/COLING '98)*. Association for Computational Linguistics, USA, 79–85. https://doi.org/10.3115/980845.980859

[3] Albert-László Barabási and Márton Pósfai. 2016. *Network science.* Cambridge University Press, Cambridge. http://barabasi.com/networksciencebook/

[4] Marcilio CP de Souto, André LV Coelho, Katti Faceli, Tiemi C Sakata, Viviane Bonadia, and Ivan G Costa. 2012. A comparison of external clustering evaluation

indices in the context of imbalanced data sets. In *2012 Brazilian Symposium on Neural Networks*. IEEE, IEEE Computer Society, Curitiba, Paraná, Brazil, 49–54.
[5] Filippo Menczer, Santo Fortunato, and Clayton A. Davis. 2020. *A First Course in Network Science*. Cambridge University Press, Cambridge. https://doi.org/10.1017/9781108653947
[6] Jose G. Moreno and Gaël Dias. 2015. Adapted B-CUBED Metrics to Unbalanced Datasets. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 911–914. https://doi.org/10.1145/2766462.2767836
[7] Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28, 1 (2002), 19–36.
[8] Lior Rokach. 2009. A survey of clustering algorithms. In *Data Mining and knowledge discovery handbook*. Springer, Boston, MA, 269–298.
[9] Gregor Wiedemann and Gerhard Heyer. 2021. Multi-Modal Page Stream Segmentation with Convolutional Neural Networks. *Lang. Resour. Eval.* 55, 1 (2021), 127–150. https://doi.org/10.1007/s10579-019-09476-2

A FIGURES FOR ELM CONSTRAINT PROOFS

A.1 Homogeneity Constraint

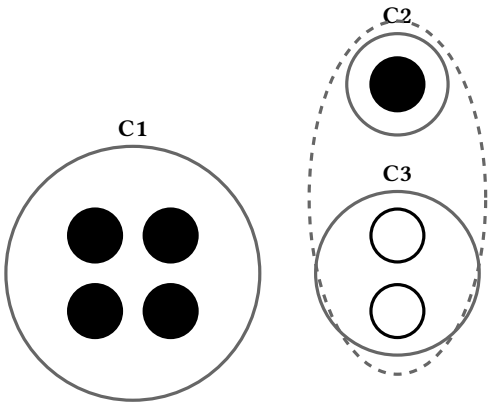

Figure 5: Homogeneity constraint: black nodes belong to one cluster and the white nodes belonging to another cluster. Shown are two partitions: the homogeneous $D_1 : \{C_1, C_2, C_3\}$ and the mixed $D_2 : \{C_1, C_2 \cup C_3\}$

.

A.2 Rag Bag Constraint

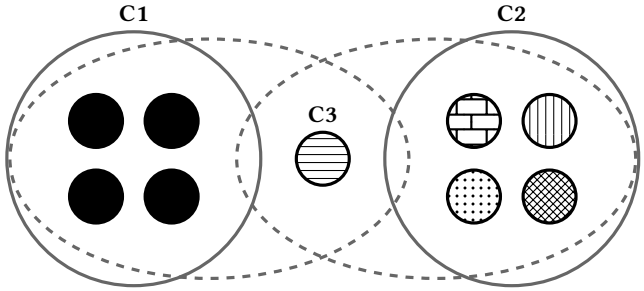

Figure 6: Rag Bag constraint: black nodes belong to one cluster and all other nodes are singleton clusters. Shown are two cluster assignments: $D_1 = \{C1, C2 \cup C3\}$ and $D_2 = \{C1 \cup C3, C2\}$.

A.3 Completeness Constraint

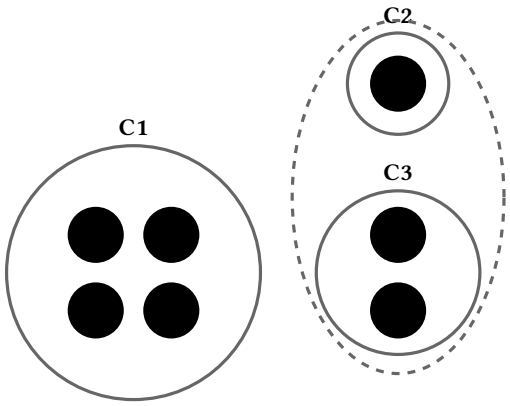

Figure 7: Completeness constraint: All nodes belong to the same cluster. Shown are two partitions: $D_1 = \{C_1, C_2 \cup C_3\}$ and $D_2 = \{C_1, C_2, C_3\}$