Adapted B-CUBED Metrics to Unbalanced Datasets

Jose G. Moreno
Normandie University
UNICAEN, GREYC CNRS
F-14032 Caen, France
jose.moreno@unicaen.fr

Gaël Dias
Normandie University
UNICAEN, GREYC CNRS
F-14032 Caen, France
gael.dias@unicaen.fr

ABSTRACT

B-CUBED metrics have recently been adopted in the evaluation of clustering results as well as in many other related tasks. However, this family of metrics is not well adapted when datasets are unbalanced. This issue is extremely frequent in Web results, where classes are distributed following a strong unbalanced pattern. In this paper, we present a modified version of B-CUBED metrics to overcome this situation. Results in toy and real datasets indicate that the proposed adaptation correctly considers the particularities of unbalanced cases.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information search and retrieval—*clustering*

Keywords

Evaluation, Search results clustering, Unbalanced datasets

1. INTRODUCTION

Evaluation of partitions obtained as a result of clustering algorithms is a challenging task. Two main kinds of metrics can be identified: supervised and unsupervised metrics. In this paper, we will deal with the former. In the information retrieval area, a recent study proposed the use of a family of metrics known as B-CUBED [1], which is used when clusters of documents are evaluated. These new metrics successfully satisfy a set of formal constraints that include problematic situations such as *Cluster Homogeneity*, *Cluster completeness*, *Rag Bag*, and finally, *Cluster size vs. quantity*. Each of these constraints evaluate a different situation that must be solved with a good evaluation metric. However, in the particular case of unbalanced datasets, these metrics fail to identify the correct solution [4]. The particularity of an unbalanced dataset is that one of the classes covers most of the document collection. Namely, this is the case when the set of documents to be clustered is dominated by one class, e.g., one of the classes covers a high percentage of documents and the remaining documents belong to many small classes. This is not a strange situation. Indeed, this is a recurrent case when the Web Search Results Clustering (SRC) problem is studied. SRC consists in grouping Web results in meaningful clusters where each cluster should "hopefully" correspond to a unique topic. Moreover, it is often the case that topics are not equally distributed in Web results. For example, consider the results obtained with a search engine and presented in Figure 1. Note that mainly two topics can be found in the results, the animal and the car. The total number of images related to the animal is almost 5 times the number of images related to the car[1]. This example clearly illustrates the existence of unbalance between the two classes[2]. In general, this behaviour can be observed in several Web SRC datasets including ODP239 [3], MORESQUE [7] and WEBSRC401 [6]. For this reason, the use of clustering evaluation metrics must be verified in unbalanced cases. This is recurrently present in the SRC problem as well as in other clustering problems.

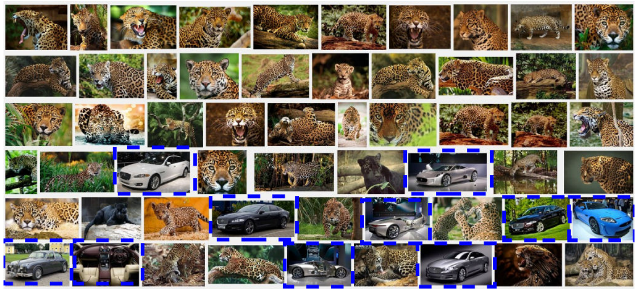


Figure 1: Commercial search engine results for the query "jaguar". 57 Web image results visualized, 47 of which are related to the animal and only 10 to the car.

In this paper, we present an evaluation of the B-CUBED metrics family using SRC datasets. Our results support the idea that B-CUBED give high scores to algorithms that follow similar distributions to the topics and otherwise, low scores even when cluster are randomly assigned. This can be explained by saying that B-CUBED metrics were also designed to penalize the erroneous links created between two classes more than putting documents in the wrong class [2]. Finally, we show how B-CUBED metrics can be modified to consider the evaluation of datasets that present the unbalanced issue. The remainder of this paper includes a description of B-CUBED clustering metrics and their modifications in Section 2. Experiments and results are presented in Section 3 and finally, discussion and conclusions are presented in Sections 4 and 5.

SIGIR'15 August 09 - 13, 2015, Santiago, Chile
©2015 ACM ISBN 978-1-4503-3621-5/15/08 ...$15.00
DOI: http://dx.doi.org/10.1145/2766462.2767836.

[1] Surrounded by the dotted blue rectangle.

[2] Many reasons could explain this distribution, however, how it affects user interaction with Web results is out of the scope of this paper.

2. ADAPTED B-CUBED FOR SRC

SRC algorithms have been evaluated with several supervised and unsupervised clustering metrics. In the former category, B-CUBED metrics have received a lot of attention in recent years. Similarly, SRC has also privileged these metrics but their impact in this particular problem is not clearly discussed. The particularities of the SRC problem motivate our efforts to develop an adapted version of these metrics.

2.1 B-CUBED metrics

B-CUBED metrics were originally proposed in [2], but exhaustively studied in [1] where it is shown that they can successfully evaluate partitions in situations included in defined formal constraints. Full comparison with illustrated examples can be found in [1]. B-CUBED F-measure (F_{b^3}), Precision (P_{b^3}) and Recall (R_{b^3}) are defined in Equations 1, 2 and 3.

$$\frac{1}{F_{b^3}} = \frac{\alpha}{P_{b^3}} + \frac{1-\alpha}{R_{b^3}} \qquad (1)$$

$$\begin{aligned} P_{b^3} &= \frac{1}{N}\sum_{i=1}^{k}\frac{1}{|\pi_i|}\sum_{x_j\in\pi_i}\sum_{x_l\in\pi_i} g_0^*(x_j,x_l) \\ R_{b^3} &= \frac{1}{N}\sum_{i=1}^{k}\frac{1}{|\pi_i^*|}\sum_{x_j\in\pi_i^*}\sum_{x_l\in\pi_i^*} g_0(x_j,x_l) \end{aligned} \qquad (2)$$

$$\begin{aligned} g_0(x_i,x_j) &= \begin{cases} 1 \iff \exists l : x_i \in \pi_l \wedge x_j \in \pi_l \\ 0, otherwise \end{cases} \\ g_0^*(x_i,x_j) &= \begin{cases} 1 \iff \exists l : x_i \in \pi_l^* \wedge x_j \in \pi_l^* \\ 0, otherwise \end{cases} \end{aligned} \qquad (3)$$

where π_i is the cluster solution i and π_i^* is the gold standard of the category i and N is the total number of documents.

2.2 Adapted B-CUBED metrics

Two main parameters of B-CUBED metrics can be modified. First, the α parameter in Equation 1 can vary to alter the importance of P_{b^3} and R_{b^3}. This issue will be discussed in section 3. Second, the number of elements considered to calculate the Precision or Recall, i.e., the number of inputs received by $g_0(\cdot,\cdot)$ and $g_0^*(\cdot,\cdot)$ can be extended to three or more elements. The new formulation to allow the use of several elements[3] is presented in Equation 4.

$$\begin{aligned} g_0(\vec{x}) &= \begin{cases} 1 \iff \exists l : \forall x_i \in \vec{x}, x_i \in \pi_l \\ 0, otherwise \end{cases} \\ g_0^*(\vec{x}) &= \begin{cases} 1 \iff \exists l : \forall x_i \in \vec{x}, x_i \in \pi_l^* \\ 0, otherwise \end{cases} \end{aligned} \qquad (4)$$

Note that because more possible combinations are considered, the normalization factors in Equation 2, $\frac{1}{|\pi_i|}$ or $\frac{1}{|\pi_i^*|}$ must be modified. After mathematical factorization, the normalization value is cancelled by the modified Precision ($P_{b^3}^{mod}$) and it is factorized in terms of R_{b^3} by the modified Recall ($R_{b^3}^{mod}$). Factorized versions of the adapted B-CUBED metrics are presented in Equation 5.

$$P_{b^3}^{mod} = P_{b^3} \wedge R_{b^3}^{mod} = R_{b^3}^{|\vec{x}|-1} \Rightarrow \frac{1}{F_{b^3}^{mod}} = \frac{\alpha}{P_{b^3}} + \frac{1-\alpha}{R_{b^3}^{|\vec{x}|-1}} \qquad (5)$$

Note that, when the number of elements considered by the g_0 functions is equal to 2, i.e. $|\vec{x}| = 2$, then the $F_{b^3}^{mod} = F_{b^3}$. In particular, the new $F_{b^3}^{mod}$ tends to give less importance to partitions with high Recall and benefits Precision preserving the α parameter.

3. EXPERIMENTS AND RESULTS

3.1 Datasets

In our experiments, we use a total of five toy examples that include four classical clustering situations as well as one situation that represents the unbalanced case. These toy examples are included in the first row of Table 1. Note that for each example, a left and right partition is included. In all cases, the right partition is considered a more adequate solution. Finally, in order to show the impact in real situations, we perform experiments in three SRC datasets: ODP239 [3], MORESQUE [7] and WEBSRC401 [6].

3.2 Clustering algorithms

SRC ALGORITHMS: A host of classical and recent algorithms were used: LINGO, STC and CascadeSRC. LINGO is based on the spectral decomposition of a term-document matrix to define the respective clusters. Finally, labels are assigned by choosing the best representative for each found cluster. STC clusters documents based on a suffix tree. Clusters are determined from the tree by selecting the longest set of strings which are used as labels. CascadeSRC [5] is a two level combination algorithm that preserves the quality in terms of intra-document similarity offered by LINGO and the compactness offered by STC.

RANDOM ALGORITHMS: Two random algorithms are studied to verify the impact of document distribution in the obtained partitions. First, the UniformRand algorithm assigns documents to each cluster in such a way that, in the end, each partition contains equally sized clusters. Secondly, the UltraShapedRand algorithm imitates the unbalanced SRC distribution. In this case, if k clusters are required, then for the clusters $c_1,..,c_{k-1}$ only one document is randomly assigned and the remaining documents are assigned to cluster c_k. Note that this distribution is an extreme case of SRC but allows to show the difference from a uniform distribution.

3.3 Formal Constraints

The formal constraints are listed as *Cluster Homogeneity*, *Cluster completeness*, *Rag Bag* and *Cluster size vs. quantity*. *Cluster Homogeneity* consists in giving a higher score to partitions where clusters contain elements of only one class, *Cluster completeness* gives higher scores to partitions where classes are represented by few clusters, *Rag Bag* gives higher scores to partitions where only one cluster contains different classes than to several clusters containing different classes. Finally, *Cluster size vs. quantity* gives higher scores to partitions where few clusters are provided but separates most classes. In addition to these formal constraints, the *Unbalanced* constraint was recently added by [4] and evaluates if a misclassification is present in a big class or in a small one. This constraint gives better scores when the incorrect classified element is from the biggest class. Results using the examples proposed by [1] and [4][4] are shown in Table 1[5]. For each example, the first column shows the value obtained with the metric for the left partition, the second column shows the result for the right partition and the third column indicates if the formal constraint is satisfied (✓) or

[3] The number of elements will be determined by the size of $\vec{x}$.

[4] The original example was slightly modified to put only one misclassified document in each evaluated partition.

[5] All metrics can be found in [7] and [1].

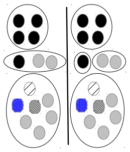
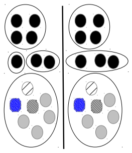
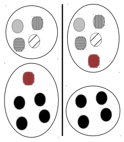
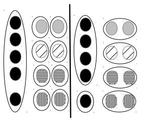
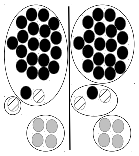
	C. Homogenity			C. Completeness			Rag Bag			C. size vs q.			Unbalanced			4 + 1 F.C.
																
Purity	0.71	0.79	✓	0.79	0.79	✗	0.56	0.56	✗	1.00	1.00	✗	0.96	0.96	✗	✖
Inv. Purity	0.79	0.79	✗	0.79	0.79	✗	1.00	1.00	✗	0.69	0.92	✓	0.96	0.96	✗	✖
F&M	0.47	0.49	✓	0.47	0.53	✓	0.61	0.61	✗	0.85	0.85	✗	0.95	0.94	✗	✖
RandIndex	0.68	0.70	✓	0.68	0.70	✓	0.72	0.72	✗	0.95	0.95	✗	0.94	0.94	✗	✖
Adj.RandIndex	0.25	0.28	✓	0.24	0.31	✓	0.40	0.40	✗	0.80	0.80	✗	0.79	0.79	✗	✖
Jaccard	0.31	0.33	✓	0.31	0.36	✓	0.38	0.38	✗	0.71	0.71	✗	0.90	0.89	✗	✖
F-measure	0.71	0.79	✓	0.79	0.79	✗	0.56	0.56	✗	1.00	1.00	✗	0.96	0.96	✗	✖
P_{b^3}	0.60	0.69	✓	0.69	0.69	✗	0.49	0.56	✓	1.00	1.00	✗	0.93	0.95	✓	✖
R_{b^3}	0.70	0.70	✗	0.71	0.76	✓	1.00	1.00	✗	0.69	0.88	✓	0.96	0.93	✗	✖
F_{b^3}	0.64	0.69	✓	0.70	0.72	✓	0.55	0.71	✓	0.82	0.93	✓	0.94	0.93	✗	✖
$P_{b^3}^{mod}$	0.60	0.69	✓	0.69	0.69	✗	0.49	0.56	✓	1.00	1.00	✗	0.93	0.95	✓	✖
$R_{b^3}^{mod}$ ($\vert\vec{x}\vert = 3$)	0.45	0.45	✗	0.56	0.57	✓	1.00	1.00	✗	0.46	0.77	✓	0.93	0.86	✗	✖
$F_{b^3}^{mod\&0.9}$	0.58	0.66	✓	0.67	0.68	✓	0.52	0.58	✓	0.90	0.97	✓	0.93	0.95	✓	✔
$F_{b^3}^{0.9}$	0.61	0.70	✓	0.69	0.70	✓	0.52	0.58	✓	0.96	0.99	✓	0.93	0.94	✓	✔

Table 1: Satisfaction of formal constraints with common SRC metrics: Examples.

not (✗). Finally, the column "4+1 F.C." indicates if the five formal constraints are satisfied simultaneously.

Note that none of current metrics can satisfy all constraints. Indeed, F_{b^3} satisfies the first 4 F.C., but misses the correct identification of the best partition for the unbalanced case as reported by [4]. However, the proposed modifications $F_{b^3}^{mod\&0.9}$ (with $|\vec{x}| = 3$) and $F_{b^3}^{0.9}$ manage to correctly classify all the formal constraints using the parameter $\alpha = 0.9$. Indeed, positive values are obtained starting from $\alpha = 0.7$, but to achieve a more general solution $\alpha = 0.9$ was selected. Our choice is motivated by the reduction of the bias generated by unbalanced datasets namely for the SRC task. It is important to remark that when $\alpha > 0.5$, Precision receives more importance than Recall.

3.4 Results in SRC datasets

A total of 10 runs were performed for each random algorithm. $F_{b^3}^{mod\&0.9}$ and $F_{b^3}^{0.9}$ average values of the two random algorithms are presented in Table 2 for different k values (from 2 to 20) and using the three SRC datasets. The UltraShapedRand algorithm behaves better than the UniformRand when evaluated with both metrics using the mentionned datasets[6]. Although when $k = 2$ both algorithms score similarly, the differences get larger as the number of k partitions grows. This was observed for both metrics in the three datasets. However, when $k = 20$, the differences are larger for $F_{b^3}^{mod\&0.9}$ than $F_{b^3}^{0.9}$. It is because $F_{b^3}^{0.9}$ gives high importance to Precision allowing to get good performance by just getting more clusters. Indeed, when the number of clusters is increasing, the number of elements by cluster must be reduced. This situation reduces the chances of putting together elements from different classes which implicitly increases Precision.

When using MORESQUE and ODP239, $F_{b^3}^{0.9}$ gives better scores to the UltraShapedRand algorithm as the number of k partitions increases. Again, this situation is given by the parameter $\alpha = 0.9$, which gives higher importance to Precision than Recall. However, this situation is not the same for $F_{b^3}^{mod\&0.9}$. This metric does not always give better scores to this situation and partitions with higher numbers of clusters may not be preferred. This is an important issue, because results suggest that $F_{b^3}^{0.9}$ will prefer partitions with clusters that contain a unique document which is not the case in any of the used datasets.

A summary of three SRC algorithms (LINGO, STC, CascadeSRC) using F_{b^3}, $F_{b^3}^{0.9}$ and $F_{b^3}^{mod\&0.9}$ is presented in Table 3. Note that for MORESQUE and ODP239, $F_{b^3}^{0.9}$ and $F_{b^3}^{mod\&0.9}$ give better scores to the SRC algorithms than to the random strategies, as it is expected for a good evaluation metric[7]. Unfortunately, the behaviour is different for WEBSRC401, where none of the metrics manages to correctly assign the scores when compared with the random strategies. However, as shown in [6], WEBSRC401 is a hard SRC dataset. But this still raises discussion.

4. DISCUSSION

Although many clustering evaluation metrics exist, none of them can consider all possible situations. Indeed, new metrics could be proposed to simultaneously deal with the formal constraints as well as adapt to the specific task. However, as shown in Table 1, this is a hard task. Moreover, we have presented $F_{b^3}^{mod\&0.9}$ which is a modified version of the B-CUBED metrics. Our proposal manages to correctly classify the examples used to validate the initial 4 formal constraints and the case for unbalanced datasets. Note that a simple α parameter modification (the $F_{b^3}^{0.9}$ metric) also manages to correctly classify the examples, but fails when it is evaluated in real datasets. It is mainly due to the fact that too much importance to Precision is given thus privileging partitions with many clusters formed by few documents. On the other hand, the $F_{b^3}^{mod\&0.9}$ not

[6] This situation was also observed for the F_{b^3} metric.

[7] This is not an evident situation. Remember that, as shown by [4], F_{b^3} can not select the correct partition in unbalanced datasets.

			k									
			2	4	6	8	10	12	14	16	18	20
$F_{b^3}^{0.9}$	MORESQUE	UniformR.	**0.3282**	0.3179	0.3103	0.3105	0.3081	0.3102	0.3097	0.3135	0.3134	0.3184
		UltraSh.R.	0.3483	0.3586	0.3706	0.3814	0.3940	0.4028	**0.4049**	0.4048	0.3992	0.3812
	ODP239	UniformR.	0.2534	0.2567	0.2617	0.2702	0.2752	0.2826	0.2908	0.2975	0.3053	**0.3112**
		UltraSh.R.	0.2601	0.2745	0.2885	0.3042	0.3162	0.3276	0.3383	0.3448	**0.3497**	0.3464
	WEBSRC401	UniformR.	**0.5921**	0.5422	0.5097	0.4850	0.4574	0.4451	0.4330	0.4208	0.4139	0.3994
		UltraSh.R.	**0.6453**	0.6393	0.6407	0.6381	0.6472	0.6380	0.6402	0.6393	0.6438	0.6432
$F_{b^3}^{mod\&0.9}$	MORESQUE	UniformR.	**0.2700**	0.1672	0.1248	0.1138	0.1096	0.1095	0.1105	0.1113	0.1079	0.1114
		UltraSh.R.	0.3479	0.3568	0.3666	0.3727	**0.3775**	0.3724	0.3489	0.3154	0.2590	0.1890
	ODP239	UniformR.	**0.2298**	0.1721	0.1355	0.1129	0.0972	0.0894	0.0798	0.0747	0.0683	0.0638
		UltraSh.R.	0.2599	0.2739	0.2870	0.3005	0.3091	**0.3134**	0.3119	0.2947	0.2573	0.1940
	WEBSRC401	UniformR.	**0.4614**	0.2144	0.1350	0.1079	0.0871	0.0782	0.0875	0.0722	0.0702	0.0649
		UltraSh.R.	0.6108	0.6040	0.6053	0.6029	**0.6119**	0.6030	0.6048	0.6039	0.6079	0.6081

Table 2: $F_{b^3}^{mod\&0.9}$ and $F_{b^3}^{0.9}$ results for partitions obtained with different k clusters of the UltraShapedRandom (UltraSh.R.) and the UniformRandom (UniformR.) algorithms using real datasets. In bold the best score for each random algorithm.

		F_{b^3}	$F_{b^3}^{0.9}$	$F_{b^3}^{mod\&0.9}$
MORESQUE	STC	**0.4602**	0.5715	**0.4186**
	LINGO	0.3989	**0.5784**	0.3497
	CascadeSRC	0.4602	0.4386	0.3874
ODP239	STC	0.4027	0.4369	**0.3410**
	LINGO	0.3461	**0.5162**	0.2902
	CascadeSRC	**0.4229**	0.3463	0.3303
WEBSRC401	STC	0.4293	0.6135	0.3618
	LINGO	0.3095	0.5758	0.2279
	CascadeSRC	**0.6665**	**0.6349**	**0.5955**

Table 3: F_{b^3}, $F_{b^3}^{0.9}$ and $F_{b^3}^{mod\&0.9}$ results for partitions obtained with STC, LINGO and CascadeSRC using real datasets. In bold the best score by metric and dataset.

only deals with the formal constraints but is also not disoriented by the random algorithms. Note from Table 2 that for the UniformRandom, $F_{b^3}^{mod\&0.9}$ reduces the assigned score as the number of clusters increases. On contrary, for the UltraShapedRand, it increases until a certain point from which it starts to decrease. These behaviours were observed for both algorithms in the three datasets.

These results could inspire the development of (1) new analysis to identify more cases (such as the unbalanced) that must be considered in the SRC problem, (2) new metrics or adaptations of existing ones to satisfy the 4+1 studied formal constraints and (3) SRC strategies that consider adapted optimization functions to obtain the satisfaction of the formal constraints. Regarding the last one, some of the existing algorithms implicitly capture these characteristics, i.e., classical SRC algorithms, such as LINGO and STC, generate shapes similar to UltraShapedRand without explicitly including it in their algorithm. This situation could explain why these are hard to beat algorithms. Indeed, the classical K-means algorithm generates partition shapes similar to the UniformRand algorithm and usually its performance is under what is obtained with LINGO or STC.

5. CONCLUSIONS

This paper presents a study about B-CUBED metrics and proposes an non-trivial adaptation of the F_{b^3} to be used in the SRC problem. Unbalanced datasets are implicitly used in the SRC problem and it is a frequently ignored issue in recent studies. Several experiments were performed in toy examples and real datasets. Main findings indicate that our proposed metric ($F_{b^3}^{mod\&0.9}$ with $|\vec{x}| = 3$) is the only one to correctly classify the toy examples in the evaluation of the formal constraints including the unbalanced case, and at the same time, able to give adequate scores when comparing SRC algorithms against random algorithms with unbalanced shapes. New research in SRC must consider the effect of using unbalanced datasets by using adapted metrics to achieved more adequate results. Similarly, existing metrics based on F_{b^3} must reconsider the unbalanced effect in the datasets. Our immediate work consists in the exploration of bigger sizes for $|\vec{x}|$ that will help in the understanding of this parameter.

6. REFERENCES

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

[2] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics - Volume 1*, ACL '98, pages 79–85, 1998.

[3] C. Carpineto and G. Romano. Optimal meta search results clustering. In *33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 170–177, 2010.

[4] M. C. P. de Souto, A. L. V. Coelho, K. Faceli, T. C. Sakata, V. Bonadia, and I. G. Costa. A comparison of external clustering evaluation indices in the context of imbalanced data sets. In *Proceedings of the 2012 Brazilian Symposium on Neural Networks (BSNN)*, SBRN '12, pages 49–54, 2012.

[5] J. G. Moreno and G. Dias. Easy web search results clustering: When baselines can reach state-of-the-art algorithms. 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 1–5, 2014.

[6] J. G. Moreno, G. Dias, and G. Cleuziou. Query log driven web search results clustering. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 777–786, 2014.

[7] R. Navigli and D. Vannella. Semeval-2013 task 11: Word sense induction & disambiguation within an end-user application. In *Proceedings of the International Workshop on Semantic Evaluation (SEMEVAL)*, pages 1–9, 2013.