Inf Retrieval (2009) 12:461–486
DOI 10.1007/s10791-008-9066-8

A comparison of extrinsic clustering evaluation metrics based on formal constraints

Enrique Amigó · Julio Gonzalo · Javier Artiles · Felisa Verdejo

Received: 16 January 2008 / Accepted: 9 July 2008 / Published online: 28 July 2008

Abstract There is a wide set of evaluation metrics available to compare the quality of text clustering algorithms. In this article, we define a few intuitive formal constraints on such metrics which shed light on which aspects of the quality of a clustering are captured by different metric families. These formal constraints are validated in an experiment involving human assessments, and compared with other constraints proposed in the literature. Our analysis of a wide range of metrics shows that only *BCubed* satisfies all formal constraints. We also extend the analysis to the problem of overlapping clustering, where items can simultaneously belong to more than one cluster. As Bcubed cannot be directly applied to this task, we propose a modified version of Bcubed that avoids the problems found with other metrics.

Keywords Clustering · Evaluation metrics · Formal constraints

1 Motivation

The clustering task consists of grouping together those objects which are similar while separating those which are not. The difference with classification tasks is that the set of categories (or clusters) is not known a priori.

Given a similarity metric between objects, evaluation metrics can be intrinsic, i.e., based on how close elements from one cluster are to each other, and how distant from elements in other clusters. Extrinsic metrics, on the other hand, are based on comparisons between the

E. Amigó (✉) · J. Gonzalo · J. Artiles · F. Verdejo
Departamento de Lenguajes y Sistemas Informáticos, UNED, Madrid, Spain
e-mail: enrique@lsi.uned.es

J. Gonzalo
e-mail: julio@lsi.uned.es

J. Artiles
e-mail: javart@bec.uned.es

F. Verdejo
e-mail: felisa@lsi.uned.es

output of the clustering system and a *gold standard* usually built using human assessors. In this work we will focus on extrinsic measures, which are the most commonly used in text clustering problems.

When doing extrinsic evaluation, determining the distance between both clustering solutions (the system output and the gold standard) is non-trivial and still subject to discussion. Many different evaluation metrics (reviewed later in this paper) have been proposed, such as Purity and Inverse Purity (usually combined via Van Rijsbergen's F measure), Clusters and class entropy, VI measure, Q_0, V-measure, Rand Statistic, Jaccard Coefficient, Mutual Information, etc.

There have already been some attempts to analyze and compare the properties of the different metrics available. Strehl (2002) compares several metrics according to their different biases and scaling properties: purity and entropy are extreme cases where the bias is towards small clusters, because they reach a maximal value when all clusters are of size one. Combining precision and recall via a balanced F measure, on the other hand, favors coarser clusterings, and random clusterings do not receive zero values (which is a scaling problem). Finally, according to Strehl' study, Mutual Information has the best properties, because it is unbiased and symmetric in terms of the cluster distribution and the gold-standard. This kind of information is very helpful to determine which metric to use in a specific clustering scenario.

Our goal is to perform a similar study, but focusing on a set of mathematical constraints that an ideal metric should satisfy. Closely related to our work is Meila (2003), where a specific metric based on entropy is tested against 12 mathematical constraints. The immediate question is why 12 constraints, or why precisely those set. In this article we also start by defining properties/constraints that any clustering metric should satisfy, but trying to observe a number of rules:

1. Constraints should be intuitive and clarify the limitations of each metric. This should allow the system developer to identify which constraints must be considered for the specific task at hand.
2. It should be possible to prove formally which metrics satisfy which properties (some previously proposed constraints can only be checked empirically).
3. The constraints should discriminate metric families, grouped according to their mathematical foundations, pointing the limitations of each metric family rather than individual metric variants. This analysis is useful for metric developers, since it ensures that further work on a certain kind of metrics will not help solving certain constraints.

We have found four basic formal constraints for clustering evaluation metrics that satisfy the above requisites. These set of constraints covers all quality aspects that have been proposed in previous work, and have been validated in an experiment involving human assessments.

Once the formal conditions have been defined and validated, we have checked all major evaluation metrics, finding that metrics from the same family behave likewise according to these formal constrains. In particular, we found BCubed metrics (BCubed precision and BCubed recall) to be the only ones that satisfy all our proposed constraints. Our work opens the possibility, however, of choosing other metrics when, for a particular clustering task, some of the restrictions do not hold, and other metric can be found to be best suited according to other criteria, such as for instance its ability to scale.

We also extend the analysis to the problem of overlapping clustering, proposing an extension of BCubed metrics which satisfies all our formal requirements.

Finally, we examine a case of study in which the combination of (extended) BCubed metrics is compared with the most commonly used pair of metrics, Purity and Inverse Purity.

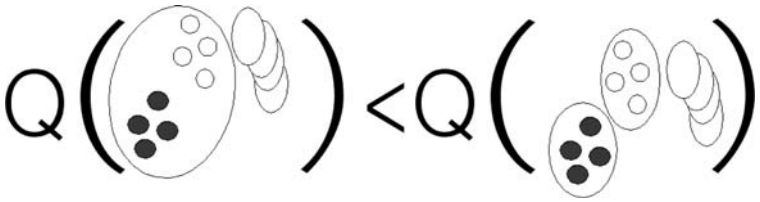


Fig. 1 Constraint 1: cluster homogeneity

The case of study shows that, unlike Purity and Inverse Purity, the proposed combination is able to discriminate and penalize an undesirable, "cheat" clustering solution.

The remainder of the paper is structured as follows: In Sect. 2, we introduce and discuss the set of proposed formal constraints. In Sect. 3, we describe the experimental procedure to validate the constraints, and discuss its results. In Sect. 4, we analyze current metrics according to our proposed constraints. Then, in Sect. 5, we compare our formal constraints with previously proposed constraint sets in the literature. In Sect. 6, we address the evaluation of overlapping clustering and propose and extended version of BCubed metrics to handle the problem adequately. Our proposal is finally tested using a case of study in Sect. 6.4, and Sect. 7 ends with the main conclusions of our study.

2 Formal constraints on evaluation metrics for clustering tasks

In order to define formal restrictions on any suitable metric, we will employ the following methodology: each formal restriction consists of a pattern (D_1,D_2) of system output pairs, where D_2 is assumed to be a better clustering option than D_1 according to our intuition. The restriction on any metric Q is then $Q(D_1) < Q(D_2)$. We have identified four basic constraints which are discussed below.

2.1 Constraint 1: cluster homogeneity

This is an essential quality property that has already been proposed in previous research (Rosenberg and Hirschberg 2007). Here, we formalize it as follows: let S be a set of items belonging to categories $L_1 \ldots L_n$. Let D_1 be a cluster distribution with one cluster C containing items from two categories L_i, L_j. Let D_2 be a distribution identical to D_1, except for the fact that the cluster C is split into two clusters containing the items with category L_i and the items with category L_j, respectively. Then an evaluation metric Q must satisfy $Q(D_1) < Q(D_2)$.

This constraint is illustrated in Fig. 1; it is a very basic restriction which states that the clusters must be homogeneous, i.e. they should not mix items belonging to different categories.

2.2 Constraint 2: cluster completeness

The counterpart to the first constraint is that items belonging to the same category should be grouped in the same cluster.[1] In other words, different clusters should contain items

[1] As in Rosenberg and Hirschberg (2007), we use the term "Completeness" to avoid "Compactness", which in the clustering literature is used as an internal property of clusters which refers to minimizing the distance between the items of a cluster.

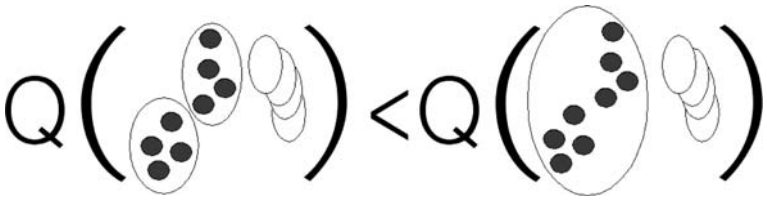


Fig. 2 Constraint 2: cluster completeness

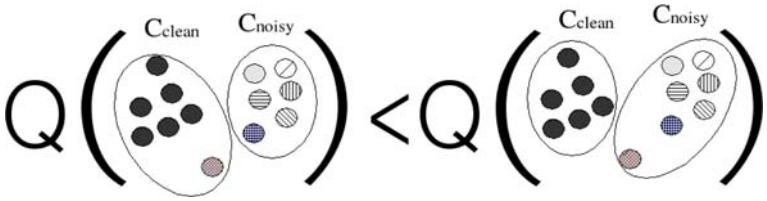


Fig. 3 Constraint 3: rag bag

from different categories. We can model this notion with the following formal constraint: Let D_1 be a distribution such that two clusters C_1,C_2 only contain items belonging to the same category L. Let D_2 be an identical distribution, except for the fact that C_1 and C_2 are merged into a single cluster. Then D_2 is a better distribution: $Q(D_1) < Q(D_2)$. This restriction is illustrated in Fig. 2.

Constraints 1 and 2 are the most basic restrictions that any evaluation metric must hold and refer to the basic goals of a clustering system: keeping items from the same category together, and keeping items from different categories apart. In the next section we will see that, surprisingly, some of the most popular metrics fail to satisfy these constraints.

2.3 Constraint 3: rag bag

An additional intuition on the clustering task is that introducing disorder into a disordered cluster is less harmful than introducing disorder into a clean cluster. Indeed, for many practical situations it is useful to have a "rag bag" of items which cannot be grouped with other items (think of "miscellaneous", "other", "unclassified" categories); it is then assumed that such a set contains items of diverse genre. Of course, in any case a perfect clustering system should identify that these items cannot be grouped and belong to different categories. But when comparing sub-optimal solutions, the intuition is that it is preferable to have clean sets plus a "rag bag" than having sets with a dominant category plus additional noise.

The boundary condition, which makes our third restriction, can be stated as follows: Let C_{clean} be a cluster with n items belonging to the same category. Let C_{noisy} be a cluster merging n items from unary categories (there exists just one sample for each category). Let D_1 be a distribution with a new item from a new category merged with the highly clean cluster C_{clean}, and D_2 another distribution with this new item merged with the highly noisy cluster C_{noisy}. Then $Q(D_1) < Q(D_2)$ (see Fig. 3). In the next section we will see that this constraint is almost unanimously validated by our human judges via examples.

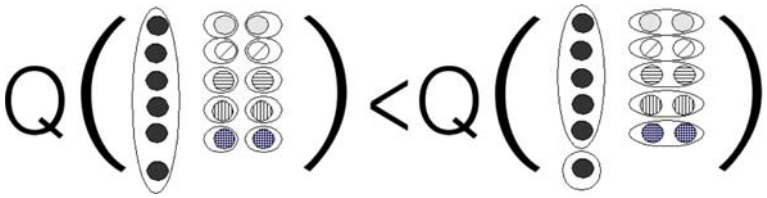


Fig. 4 Clusters size versus quantity

2.4 Constraint 4: clusters size versus quantity

A small error in a big cluster should be preferable to a large number of small errors in small clusters. This property is partially related with the fourth property in Meila (2003), called in Rosenberg and Hirschberg (2007) as *n-invariance*. We state a boundary condition related to this notion saying that separating one item from its class of $n > 2$ members is preferable to fragmenting n binary categories (see Fig. 4).

Formally, let us consider a distribution D containing a cluster C_l with $n + 1$ items belonging to the same category L, and n additional clusters $C_1 \ldots C_n$, each of them containing two items from the same category $L_1 \ldots L_n$. If D_1 is a new distribution similar to D where each C_i is split in two unary clusters, and D_2 is a distribution similar to D, where C_l is split in one cluster of size n and one cluster of size 1, then $Q(D_1) < Q(D_2)$.

3 Testing the formal constraints

We now want to test whether our formal constraints reflect common intuitions on the quality of a clustering. For this, we have performed an experiment in which we presented pairs of alternative clustering options to eight human assessors, and they were asked to select the best option in each pair. Every pair was designed to match one of the constraints, so that each assessor's choice confirms or contradicts the constraint.

We have used the EFE 1994–1995 CLEF corpus (Gonzalo and Peters 2005) to generate the test set. This corpus consists of news-wire documents in Spanish, along with a set of topics and relevance judgments for each of the topics. We have randomly selected six queries and ten relevant documents per query, and then we have used the documents for each query as a category. Note (Fig. 9) that each piece of news is manually tagged with a rather specific keyword description, which makes the clustering task easier to the assessors. Titles for the selected topics were "UN forces in Bosnia", "Invasion of Haiti", "War in Chechnya", "Uprising in Chiapas", "Operation Turquoise in Ruanda" and "Negotiations in Middle East".

For each formal constraint, we have implemented an algorithm which randomly generates pairs of two distributions which are instances of D_1 and D_2:

- *Cluster homogeneity* (see Fig. 5) (1) We generate three clusters C_1, C_2 and C_3 containing titles from a topic L_{13} (the subscript 13 indicating that there are items from this topic in clusters C_1 and C_3), and from another topic L_2 (which has items in C_2) such that $|C_1| + |C_2| < |C_3|$. (2) We generate a cluster C_4 containing news titles from several random topics, such that most of them correspond to one single topic L' different from L_{13} and L_2. (3) Then we build the following distributions:

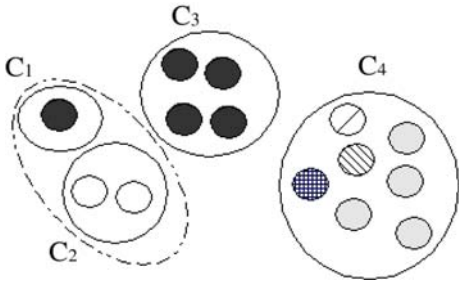


Fig. 5 Example of test to validate the cluster homogeneity constraint

$$D_1 = \{C_1 \cup C_2, C_3, C_4\}$$
$$D_2 = \{C_1, C_2, C_3, C_4\}$$

- *Cluster completeness* (see Fig. 6) (1) We generate three clusters C_1, C_2 and C_3 containing titles from the same topic L, with $|C_1| + |C_2| < |C_3|$. (2) The cluster C_4 is generated as in the previous algorithm. (3) Then we build the following distributions:

$$D_1 = \{C_1, C_2, C_3, C_4\}$$
$$D_2 = \{C_1 \cup C_2, C_3, C_4\}$$

- *Rag bag* (see Fig. 7) (1) We generate a cluster C_1 with four titles, each from a different topic. (2) We generate a cluster C_2 with four titles from the same topic. (3) We generate a cluster C_3 with one title from a new topic. (4) We compare the distributions:

$$D_1 = \{C_1, C_2 \cup C_3\}$$
$$D_2 = \{C_1 \cup C_3, C_2\}$$

- *Cluster size versus quantity* (see Fig. 8) (1) We generate four clusters C_1, C_2, C_3 and C_4 each one with two titles from the same topic. (2) We split these clusters in two C_i' and C_i''. (3) We generate a cluster C_5 with five titles from the same topic. (4) We extract one item from C_5 generating C_i' and C_i''. (5) We compare the distributions:

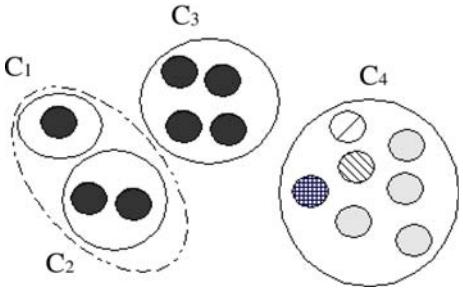


Fig. 6 Example of test to validate the cluster completeness constraint

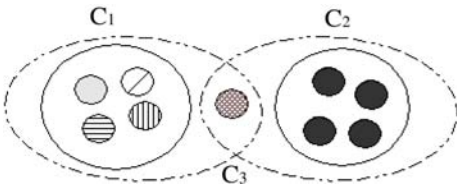


Fig. 7 Example of test to validate the Rag Bag constraint

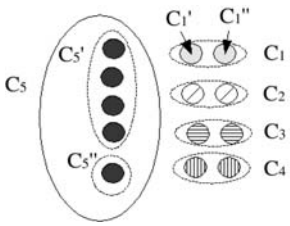


Fig. 8 Sample of distribution to validate the cluster size versus quality constraint

$$D_1 = \{C'_1, C''_1, C'_2, C''_2, C'_3, C''_3, C'_4, C''_4, C_5\}$$
$$D_2 = \{C_1, C_2, C_3, C_4, C'_5, C''_5\}$$

Eight volunteers and five instances per constraint have been employed in this experiment, for a total of 40 individual assessments. For each instance, both distributions were presented to the volunteers, showing only the titles of the documents. The instructions asked the assessors to decide if the first distribution was better, worse or roughly equivalent to the second one. The ordering of both distributions (D_1 and D_2) and the titles within each cluster have been randomly reordered for each case. Figure 9 shows an example of how the document clusters were presented to the judges.

Table 1 shows the results of the experiment. All restrictions were validated by more than 90% of the assessments. Constraint 4 was validated in all cases, and constraints 1–3 were only contradicted in one case each. Given the test conditions and the fact that eight different assessors participated in the experiment, we take these figures as a strong empirical support for the potential relevance of constraints. Note that constraints 3 and 4 (which are less obvious and more restricted in scope than constraints 1 and 2) receive even higher support than the first two constraints.

CASE 19 OPTION A

```
GROUP 1:
BOSNIA-FEDERACION: MUSULMANES PIDEN ARMAS PARA ELLOS Y MAS BLOQUEO CONTRA SERBIOS
HAITI-OEA:  ESTADOS UNIDOS INICIA CONSULTAS PARA ENVIAR MISION A HAITI
RUSIA-CHECHENIA: CHECHENIA AMENAZA GUERRA TOTAL EN CASO INVASION RUSIA
ESPAÑA-MEXICO: PIDEN A SALINAS DE GORTARI SOLUCION JUSTA CUESTION INDIGENA
ISRAEL-JORDANIA: ISRAEL CALMARA LAS PREOCUPACIONES DEL REY HUSEIN

GROUP 2:
RUANDA-GUERRA-FRANCIA: INTERVENCION FRANCESA LIMITADA HASTA FINALES DE JULIO
RUANDA-GUERRA-FRANCIA: BALLADUR: INTERVENCION FRANCESA SOLO HASTA FINAL DE JULIO
RUANDA-GUERRA-FRANCIA: TODO LISTO PARA COMIENZO OPERACION TURQUESA
RUANDA-ONU:BELGICA REITERA QUE NO ENVIARA TROPAS
```

CASE 19 OPTION B

```
GROUP 1:
ESPAÑA-MEXICO: PIDEN A SALINAS DE GORTARI SOLUCION JUSTA CUESTION INDIGENA
HAITI-OEA:  ESTADOS UNIDOS INICIA CONSULTAS PARA ENVIAR MISION A HAITI
ISRAEL-JORDANIA: ISRAEL CALMARA LAS PREOCUPACIONES DEL REY HUSEIN
RUSIA-CHECHENIA: CHECHENIA AMENAZA GUERRA TOTAL EN CASO INVASION RUSIA

GROUP 2:
BOSNIA-FEDERACION: MUSULMANES PIDEN ARMAS PARA ELLOS Y MAS BLOQUEO CONTRA SERBIOS
RUANDA-GUERRA-FRANCIA: BALLADUR: INTERVENCION FRANCESA SOLO HASTA FINAL DE JULIO
RUANDA-ONU:BELGICA REITERA QUE NO ENVIARA TROPAS
RUANDA-GUERRA-FRANCIA: INTERVENCION FRANCESA LIMITADA HASTA FINALES DE JULIO
RUANDA-GUERRA-FRANCIA: TODO LISTO PARA COMIENZO OPERACION TURQUESA
```

Fig. 9 Example of test presented to users for the rag bag constraint

Table 1 Validation of constraints by assessors: experimental results

Constraint	Validated	Contradicted	Indifferent
Cluster homogeneity	37 (92%)	1 (2.5%)	2 (5%)
Cluster completeness	36 (90%)	1 (2.5%)	3 (7.5%)
Rag bag	38 (95%)	1 (2.5%)	1 (2.5%)
Cluster size versus quantity	40 (100%)	0	0

4 Comparison of evaluation metrics

Given the large number of metrics proposed for the clustering task, we will group them in four families and try to test properties inherent to the kind of information that each family uses.

4.1 Evaluation by set matching

This metric family was identified as such in Meila (2003). They share the feature of assuming a one to one mapping between clusters and categories, and they rely on the precision and recall concepts inherited from Information Retrieval.

The most popular measures for cluster evaluation are Purity, Inverse Purity and their harmonic mean (F measure). Purity (Zhao and Karypis 2001) focuses on the frequency of the most common category into each cluster. Being C the set of clusters to be evaluated, L the set of categories (reference distribution) and N the number of clustered items, Purity is computed by taking the weighted average of maximal precision values:

$$\text{Purity} = \sum_i \frac{|C_i|}{N} \max_j \quad \text{Precision}(C_i, L_j)$$

where the precision of a cluster C_i for a given category L_j is defined as:

$$\text{Precision}(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

Purity penalizes the noise in a cluster, but it does not reward grouping items from the same category together; if we simply make one cluster per item, we reach trivially a maximum purity value. Inverse Purity focuses on the cluster with maximum recall for each category. Inverse Purity is defined as:

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{N} \max_j \quad \text{Precision}(L_i, C_j)$$

Inverse Purity rewards grouping items together, but it does not penalize mixing items from different categories; we can reach a maximum value for Inverse purity by making a single cluster with all items.

A more robust metric can be obtained by combining the concepts of Purity and Inverse Purity, matching each category with the cluster that has a highest combined precision and recall, using Van Rijsbergen's F measure (Van Rijsbergen 1974; Larsen and Aone 1999; Steinbach et al. 2000):

$$F = \sum_i \frac{|L_i|}{N} \max_j \{F(L_i, C_j)\}$$

where

$$F(L_i, C_j) = \frac{2 \times \text{Recall}(L_i, C_j) \times \text{Precision}(L_i, C_j)}{\text{Recall}(L_i, C_j) + \text{Precision}(L_i, C_j)}$$

$$\text{Recall}(L, C) = \text{Precision}(C, L)$$

One common problem with these type of metrics is that they cannot satisfy constraint 2 (cluster completeness): as each category is judged only by the cluster which has more items belonging to it, changes in other clusters are not detected. This problem has been previously identified (see Meila 2003 or Rosenberg and Hirschberg 2007). An example can be seen in Fig. 6: clusters C_1 and C_2 contain items from the same category, so merging them should improve the quality of the distribution (Category completeness constraint). But Purity does not satisfy this constraint in general, and both Inverse Purity and F measure are not sensible to this case, as the cluster with maximal precision and F measure over the category of black circles is C_3.

Figure 11 shows the results of computing several metrics in four test cases instantiating the four constraints; there, we can see counterexamples showing that no metric in this family satisfies constraints 2 and 3, and even constraint 1 is only satisfied by the Purity measure.

4.2 Metrics based on counting pairs

Another approach to define evaluation metrics for clustering is considering statistics over pairs of items (Halkidi et al. 2001; Meila 2003). Let SS be the number of pairs of items belonging to the same cluster and category; SD the number of pairs belonging to the same cluster and different category; DS the number of pairs belonging to different cluster and the same category, and DD the number of pairs belonging to different category and cluster. SS and DD are "good choices", and DS, SD are "bad choices".

Some of the metrics using these figures are:

$$\text{Rand statistic R} = \frac{(SS + DD)}{SS + SD + DS + DD}$$

$$\text{Jaccard Coefficient J} = \frac{SS}{SS + SD + DS}$$

$$\text{Folkes and Mallows FM} = \sqrt{\frac{SS}{SS + SD} \frac{SS}{SS + DS}}$$

It is easy to see that these type of metrics satisfy the first two constraints; but they do not satisfy constraints 3 and 4; Fig. 11 shows counterexamples. Take for instance the example for constraint 4: With regard to the ideal clustering, in both distributions some elements from the same category are moved apart, producing a SS decrase and a DS increase. The number of pairs affected by the fragmentation in both distributions is the same. In the first case, one black item is separated from the other four black items. In the second case, four correct binary clusters are fragmented into unary clusters. Therefore, the values for SS (10), and DS (4) are the same in both distributions. The problem is that the number of item

pairs in a cluster has a quadratic dependence with the cluster size, and then changes in bigger clusters have an excessive impact in this type of measures.

4.3 Metrics based on entropy

The entropy of a cluster (Steinbach et al. 2000; Ghosh 2003) reflects how the members of the k categories are distributed within each cluster; the global quality measure is again computed by averaging the entropy of all clusters:

$$\text{Entropy} = -\sum_j \frac{n_j}{n} \sum_i P(i,j) \times \log_2 P(i,j)$$

being $P(i, j)$ the probability of finding an element from the category i in the cluster j, n_j the number of items in cluster j and n the total number of items in the distribution. Other metrics based on entropy have also been defined, for instance, "class entropy" (Bakus et al. 2002), "variation of information" (Meila 2003) "Mutual Information" (Xu et al. 2003), Q_o (Dom 2001) or "V-measure" (Rosenberg and Hirschberg 2007).

Figure 11 shows counterexamples for some of these measures in all constraints: entropy and mutual information fail to satisfy constraints 2–4, and class entropy constraints 1 and 3. In particular, the Rag Bag constraint cannot be satisfied by any metric based on entropy: conceptually, the increase of entropy when an odd item is added is independent from the previous grade of disorder in the cluster; therefore, it is equivalent to introduce a wrong item in a clean cluster or in a noisy cluster.

Let us formalize our argument: let C be a cluster with n items. Then the entropy would be computed as

$$E_C = \sum_i P_i \log P_i$$

where P_i is the probability of finding an element of the category i in the cluster. Let C' be the same cluster adding an item that is unique in its category and was previously isolated. Then

$$E_{C'} = \frac{1}{n+1}\log\frac{1}{n+1} + \sum_i \frac{nP_i}{n+1}\log\frac{nP_i}{n+1}$$

being n the number of items in the cluster. Operating:

$$\begin{aligned} E_{C'} &= \frac{1}{n+1}\log\frac{1}{n+1} + \frac{n}{n+1}\sum_i \left[P_i * \left(\log\frac{n}{n+1} + \log P_i\right)\right] \\ &= \frac{1}{n+1}\log\frac{1}{n+1} + \frac{n}{n+1}\left[\log\frac{n}{n+1}\sum_i P_i + \sum_i P_i * \log P_i\right] \end{aligned}$$

Since $\sum_i P_i = 1$

$$E_{C'} = \frac{1}{n+1}\log\frac{1}{n+1} + \frac{n}{n+1}\left[\log\frac{n}{n+1} + E_C\right]$$

In other words, the increase in entropy depends exclusively from n; the homogeneity or heterogeneity of the cluster does not affect the result.

4.4 Evaluation metrics based on edit distance

In Pantel and Lin (2002), an evaluation metric based on transformation rules is presented, which opens a new family of metrics. The quality of a clustering distribution is related with the number of transformation rules that must be applied to obtain the ideal distribution (one cluster for each category). This set of rules includes merging two clusters and moving an item from one cluster to another. Their metric (which we do not fully reproduce here for lack of space) fails to satisfy constraints 1 and 3 (see counterexamples in Fig. 11). Indeed, metrics based on edit distance cannot satisfy the Rag Bag constraint: independently from where we introduce the noisy item, the distance edit is always one application of a transformation rule, and therefore the quality of both distributions will always be the same.

4.5 BCubed: a mixed family of metrics

We have seen that none of previous metric families satisfy all our formal restrictions. The most problematic constraints is *Rag Bag*, which is not satisfied by any of them. However, BCubed precision and recall metrics (Bagga and Baldwin 1998) satisfy all constraints. Unlike Purity or Entropy metrics, which compute independently the quality of each cluster and category, BCubed metrics decompose the evaluation process estimating the precision and recall associated to each item in the distribution. The item precision represents how many items in the same cluster belong to its category. Symmetrically, the recall associated to one item represents how many items from its category appear in its cluster. Figure 10 illustrates how the precision and recall of one item is computed by BCubed metrics.

From a user's point of view, BCubed represents the clustering system effectiveness when, after accessing one reference item, the user explores the rest of items in the cluster. If this item had a high BCubed recall, the user would find most of related items without leaving the cluster. If the reference item had a high precision, the user would not find noisy items in the same cluster. The underlying difference with Purity or Entropy measures is that the adequacy of items depends on the reference item rather than the predominant category in the cluster.

Although BCubed is defined in Bagga and Baldwin (1998) as an algorithm, it can also be described in terms of a function. Let $L(e)$ and $C(e)$ denote the category and the cluster of an item e. We can define the correctness of the relation between e and e' in the distribution as:

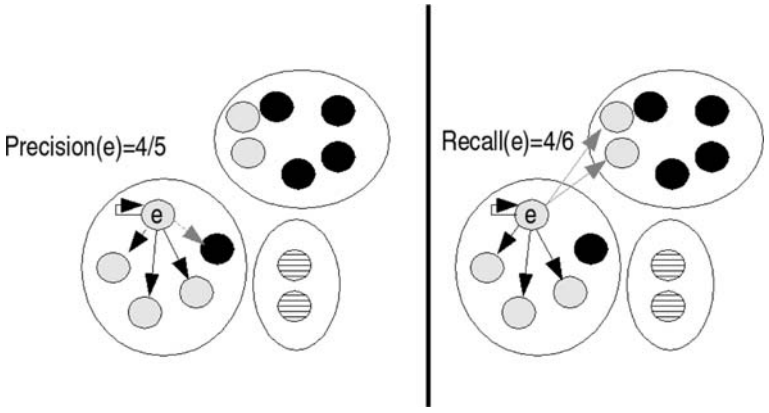


Fig. 10 Example of computing the BCubed precision and recall for one item

$$\text{Correctness}(e,e') = \begin{cases} 1 \text{ iff } L(e) = L(e') \longleftrightarrow C(e) = C(e') \\ 0 \text{ otherwise} \end{cases}$$

That is, two items are correctly related when they share a category if and only if they appear in the same cluster. BCubed precision of an item is the proportion of items in its cluster which have the item's category (including itself). The overall BCubed precision is the averaged precision of all items in the distribution. Since the average is calculated over items, it is not necessary to apply any weighting according to the size of clusters or categories. The BCubed recall is analogous, replacing "cluster" with "category". Formally:

$$\text{Precision BCubed} = \text{Avg}_e[\text{Avg}_{e'.C(e)=C(e')}[\text{Correctness}(e,e')]]$$

$$\text{Recall BCubed} = \text{Avg}_e[\text{Avg}_{e'.L(e)=L(e')}[\text{Correctness}(e,e')]]$$

BCubed combines the best features from other metric families. Just like Purity or Inverse Purity, it is inspired on precision and recall concepts, being easily interpretable. As entropy based metrics, it considers the overall disorder of each cluster, not just the predominant category, satisfying restrictions 1 and 2 (*homogeneity* and *completeness*). Both BCubed and metrics based on counting pairs consider the relation between pairs of items. However in BCubed metrics the overall average is computed over single items and the quadratic effect produced by the cluster size disappears, therefore satisfying restriction 4, *cluster size versus cluster quantity*. In addition, unlike all other metrics, BCubed also satisfies the *Rag Bag* constraint.

Let us verify the four constraints:

- *Cluster homogeneity constraint*: splitting a cluster that mixes two categories into two "pure" clusters increases the BCubed precision, and does not affect recall (see Fig. 1).
- *Cluster completeness constraint*: unifying two clusters which contain only items from the same category increases the BCubed recall measure, and the precision of joined items remains maximal (see Fig. 2).
- *Rag bag constraint*: let us suppose that we have an item (unique in its category) in an isolated cluster. Introducing the item in a clean cluster of n items (D_1, Fig. 3) decreases the precision of each item in the clean cluster from 1 to $\frac{n}{n+1}$, and the precision of the item just inserted from 1 to $\frac{1}{n+1}$. So, being N_{tot} the total number of items in the distribution, while the recall is not affected in any way, the overall precision decreasing in the distribution is:

$$\text{DEC}_{D_1} = \frac{1 + n * 1}{N_{\text{tot}}} - \frac{\frac{1}{n+1} + n * \frac{n}{n+1}}{N_{\text{tot}}} = \frac{\frac{2n}{n+1}}{N_{\text{tot}}} \simeq \frac{2}{N_{\text{tot}}}$$

On the other hand, introducing the item in a noisy cluster (D_2, Figure 3) decreases the precision of the isolated item from 1 to $\frac{1}{n+1}$, and the items in the noisy cluster from $\frac{1}{n}$ to $\frac{1}{n+1}$. So the overall decrease in the distribution is smaller:

$$\text{DEC}_{D_2} = \frac{1 + n * \frac{1}{n}}{N_{\text{tot}}} - \frac{1 * \frac{1}{n+1} + n * \frac{1}{n+1}}{N_{\text{tot}}} = \frac{1}{N_{\text{tot}}} < \text{DEC}_{D_1}$$

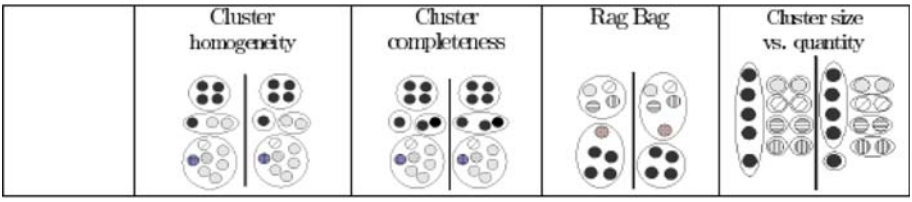
- *Cluster size versus quantity*: in the distribution D_1 from Fig. 4, $2n$ items decrease their recall in 50%. That represents an overall decrease of:

$$\mathrm{DEC}_{D_1} = \frac{2n}{N_{\mathrm{tot}}} - \frac{2n\frac{1}{2}}{N_{\mathrm{tot}}} = \frac{n}{N_{\mathrm{tot}}}$$

On the other hand, in the distribution D_2 the recall of n items decreases from 1 to $\frac{n}{n+1}$, and the recall of one item decreases from 1 to $\frac{1}{n+1}$, So the overall decrease in the distribution is smaller:

$$\mathrm{DEC}_{D_2} = \frac{n+1}{N_{\mathrm{tot}}} - \frac{n\frac{n}{n+1} + \frac{1}{n+1}}{N_{\mathrm{tot}}} = \frac{\frac{2n}{n+1}}{N_{\mathrm{tot}}} \simeq \frac{2}{N_{\mathrm{tot}}} < \mathrm{DEC}_{D_1}$$

In conclusion, BCubed metrics together satisfy all our formal constraints. BCubed precision covers restrictions 1 and 3. BCubed recall covers constraints 2 and 4. Figure 11 contains a sample of clustering distribution pair for each formal constraint. The table shows that BCubed precision and recall metrics cover all of them.



Metrics based on set matching												
Purity	**0.71**	**0.78**	√	0.78	0.78	×	0.55	0.55	×	1	1	×
Inv. Purity	0.78	0.78	×	0.78	0.78	×	1	1	×	**0.69**	**0.92**	√
F-measure	0.63	0.63	×	0.62	0.62	×	0.61	0.61	×	**0.79**	**0.96**	√
	OK			FAIL			FAIL			OK		

Metrics based on counting pairs												
Rand	**0.68**	**0.7**	√	**0.68**	**0.7**	√	0.72	0.72	×	0.95	0.95	×
Jaccard	**0.31**	**0.32**	√	**0.31**	**0.35**	√	0.37	0.37	×	0.71	0.71	×
F&M	**0.47**	**0.49**	√	**0.47**	**0.52**	√	0.61	0.61	×	0.84	0.84	×
	OK			OK			FAIL			FAIL		

Metrics based on Entropy												
-Entropy	**-1.03**	**-0.8**	√	-0.83	-0.83	×	-1.29	-1.29	×	0	0	×
-Class Entr.	-0.65	-0.65	×	**-0.69**	**-0.49**	√	0	0	×	**-0.61**	**-0.28**	√
Mutual Inf.	**0.84**	**1.03**	√	1	1	×	0.99	0.99	×	2.19	2.19	×
-VI	**-1.68**	**-1.48**	√	**-1.52**	**-1.32**	√	-1.28	-1.28	×	**-0.61**	**-0.27**	√
	OK			OK			FAIL			OK		

Metrics based on editing distance												
Edit dist.	(steps) 7	7	×	**7**	**6**	√	6	6	×	**9**	**6**	√
	FAIL			OK			FAIL			OK		

BCubed metrics												
Precision	**0.59**	**0.69**	√	0.62	0.62	×	**0.52**	**0.64**	√	1	1	×
Recall	0.69	0.69	×	**0.71**	**0.75**	√	1	1	×	**0.64**	**0.81**	√
F(BCubed)	**0.63**	**0.69**	√	**0.66**	**0.67**	√	**0.68**	**0.78**	√	**0.78**	**0.89**	√
	OK			OK			OK			OK		

Fig. 11 Satisfaction of formal constraints: examples

A remaining issue is how to combine both in a single evaluation metric. According to our formal constraints, any averaging criterion for combining metrics satisfies all formal constraints when these are satisfied by the combined metrics in isolation. This is due to the fact that our formal constraints are defined in such a way that each one represents an isolated quality aspect. When a metric does not cover a specific quality aspect, the associated restriction is not affected.

A standard way of combining metrics is Van Rijsbergen's F (Van Rijsbergen 1974) and it is computed as follows:

$$F(R,P) = \frac{1}{\alpha(\frac{1}{P}) + (1-\alpha)(\frac{1}{R})}$$

being R and P two evaluation metrics and being α and $(1-\alpha)$ the relative weight of each metric ($\alpha = 0.5$ leads to the harmonic average of P, R). The last row in Fig. 11 shows the results when applying $F_{\alpha=0.5}$ over BCubed Precision and Recall, satisfying all formal constraints.

5 Related work: other proposed formal constraints

Are four constraints enough? We do not have a formal argument supporting this, but we can at least compare our set of constraints with previous related proposals.

5.1 Dom's constraints

In Dom (2001), Dom proposes five formal constraints. These were extended to seven in Rosenberg and Hirschberg (2007). The author decomposes the clustering quality into a set of parameters: the number of "noise" and "useful" clusters, the number of "noise" and "useful" categories, and three components of the error mass probability. "Noise" clusters are those that contain items equally from each category. On the opposite, "Useful" clusters have a predominant category. The error mass probability measures to what extent single items are not included in the corresponding "useful" cluster.

The formal constraints consist of testing, over a random set of clustering samples, if specific parameter configurations do lead to a decrease of quality according to the metric. Basically, these formal constraints capture the idea that a clustering is worse when: (1) the number of useful clusters varies away from the number of categories, (2) the number of noise clusters increases and (3) the error mass parameters increase. Roughly speaking, these ideas are directly correlated with our constraints. For instance, *Cluster Homogeneity* and *Cluster Completeness* implies respectively a decrease and increase of useful clusters regarding the number of categories.

But Dom's restrictions reflect intermediate situations which are not considered explicitly by our formal constraints, since we defined them using boundary conditions. Theoretically speaking, this implies that a metric satisfying our constraints may not satisfy Dom's constraints. However, all metric drawbacks which are detected by Dom's constraints are also detected by our set.

In particular, the results in Rosenberg and Hirchberg (2007) shows that metrics based on Entropy satisfy all these formal constraints, and metrics based on counting pairs fail at least in two properties. To explain this result, the authors state that "the number of noise classes or clusters can be increased without reducing any of these metrics" when counting pairs.

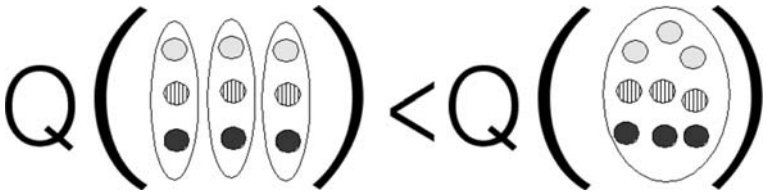


Fig. 12 More noise clusters implies less quality

We believe that our constraint 4 *Cluster size versus quantity* provides a more in-depth explanation. Increasing the number of noise clusters while fixing the rest of parameters produces smaller clusters (see Fig. 12). Metrics based on counting pairs give a quadratic relevance to erroneously joined items in bigger clusters, increasing the score when splitting noise clusters. For instance, in Fig. 12, the right distribution introduces 9 correct item associations at the expense of 27 incorrect pairs. Metrics based on entropy, on the contrary, satisfy the *Cluster size versus quantity* constraint, overcoming this problem.

Dom's constraints have some drawbacks with respect to our meta-evaluation framework:

1. Dom's constraints detect less limitations than our constraints. For instance, they do not detect drawbacks of entropy-based metrics, while they fail to satisfy our *Rag Bag* constraint.
2. Each Dom's constraint is related with several quality aspects. For instance the mass error or the number of noise clusters are related simultaneously with the concepts of *homogeneity, completeness* and *Cluster size versus quantity*. Therefore, it is not easy to identify the need for satisfying specific constraints in specific clustering applications.
3. It is not easy to prove formally that an evaluation metric satisfies Dom's constraints. Indeed, these restrictions were tested by evaluating "random" clustering distributions. Our constraints, however, can be formally verified for each family of metrics.

5.2 Meila's constraints

Meila (2003) proposes an entropy-based metric (*Variation Information* or VI) and enumerates 12 desirable properties associated with this metric. Properties 1–3, for instance, are positivity, symmetry and triangle inequality, which altogether imply that VI is a proper *metric* on clusterings. Most of these properties are not directly related to the quality aspects captured by a metric, but rather on other intrinsic features such as the ability to scale or computational cost. The most relevant properties for our discussion are:

- Property 4 is related with the *cluster size versus quantity* constraint. It states that the quality of a distribution depends on the relative sizes of clusters but not on the number of points in the data set. Metrics based on counting pairs do not satisfy this property since the number of item pairs increase quadratically regarding the number of items in the distribution.
- Property 7 states that splitting or merging smaller clusters has less impact than splitting or merging larger ones. It states also that the variation in the evaluation measure is independent of anything outside the clusters involved. Although this property is desirable, in practice all metrics discussed here satisfy it. Therefore, it does not provide much information about what metrics are more suitable for evaluation purposes.

- Properties 10 and 11 are associated to the idea that splitting all clusters according to item categories improves the results. This corresponds with the formal constraint that we call *Cluster Completeness*.

In short, while Meila's properties are an in-depth characterization of the VI metric, they do not suggest any additional constraint to our original set. Indeed, the VI metric proposed by Meila does not satisfy our constraint 3 (Rag Bag), being an entropy-based metric (see Sect. 4.3).

6 Evaluation of overlapping clustering

The metrics discussed so far do not (at least explicitly) handle clustering scenarios where the same item can be assigned to more than one cluster/category (overlapping clustering). For instance, a piece of news could be related to both "international" and "culture" sections of an electronic newspaper at the same time. Ideally, an information retrieval system based on clustering should put this article in both clusters.

This problem can be seen also as a generalization of the *hierarchical clustering* task. For instance, international news could be sub-classified into "international-culture" and "international-politics". This article would belong both to "international-culture" (child category/cluster) and "international" (parent category/cluster). From a general point of view, a hierarchical clustering is an overlapping clustering where each item that occurs in a leaf cluster occurs also in all its ancestors.

Figure 13 illustrates the relationship between hierarchical and overlapping clustering. The leftmost representation is a distribution where items 1 and 3 belong to the grey category (cluster A) and items 1 and 2 belong to the black category (cluster B). This is an overlapping clustering because item 1 belongs both to black and grey categories. The rightmost clustering is its hierarchical counterpart: the cluster A (root cluster) is associated with the grey category, and its child clusters (B and C) are associated with the categories black and white, respectively. The three items occur in the root category. In addition, items 1 and 2 belong to the left child cluster (black category) and item 3 belongs to the right child cluster (white category). In short, a hierarchical clustering is an overlapping clustering where each cluster at each level is related with a category.

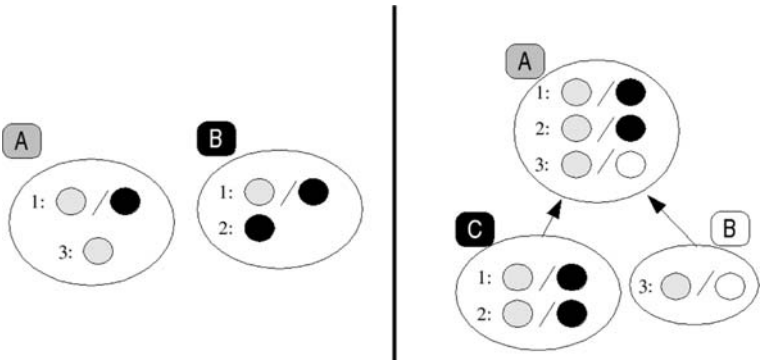


Fig. 13 Multi-category versus hierarchical clustering

6.1 Extending standard metrics for overlapping clustering

While in standard clustering each item is assigned to one cluster, in overlapping clustering each item is assigned to a set of clusters. Let us use the term "categories" to denote the set of "perfect" clusters defined in the gold standard. Then, any evaluation metric must reflect the fact that, in a perfect clustering, two items sharing n categories should share n clusters.

This apparently trivial condition is not always met. In particular, purity and entropy-based metrics cannot capture this aspect of the quality of a given clustering solution. This is because they focus on the quality of the clusters (purity) and the quality of the categories (inverse purity) independently from each other. Let us consider an example.

Figure 14 represents a clustering case where three items must be distributed hierarchically. The rightmost distribution shows the correct solution: each item (1–3) belongs to two categories and therefore appears in two clusters. The leftmost distribution, on the contrary, simply groups all items in just one cluster. This one does not represent the hierarchical structure of the correct clustering; however, the only given cluster is perfectly coherent, since all items share one category (grey). In addition, all the items from the same category share the same cluster (because there is only one). Therefore, cluster/category oriented metrics inevitably think that the leftmost cluster is perfect.

The problem with purity and inverse purity shows that the extension of quality metrics to overlapping clustering is not trivial. Addressing this problem requires another formal analysis, with a new set of formal constraints and a study of how the different metric families can satisfy them. While such a study is beyond the scope of this paper, here we will try to extend *Bcubed* metrics, which are the only ones that satisfy all formal constraints proposed in this paper, with the goal of providing a good starting point for a more in-depth study. We will show that our extension of Bcubed metrics solves some practical problems of existing metrics.

6.2 Extending BCubed metrics

BCubed metrics independently compute the precision and recall associated to each item in the distribution. The precision of one item represents the amount of items in the same cluster that belong to its category. Analogously, the recall of one item represents how many items from its category appear in its cluster.

As we stated in Sect. 4.5, the correctness of the relation between two items in a non-overlapping clustering is represented by a binary function.

$$\text{Correctness}(e, e') = \begin{cases} 1 \text{ if } L(e) = L(e') \longleftrightarrow C(e) = C(e') \\ 0 \text{ in other case} \end{cases}$$

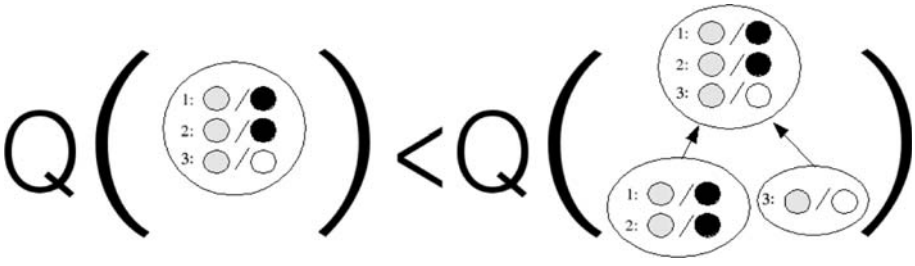


Fig. 14 Item multiplicity

where $L(e)$ is the cluster assigned to e by the clustering algorithm and $C(e)$ is the cluster assigned to e by the gold standard.

In the case of overlapping clustering the relation between two items cannot be represented as a binary function. This is due to the fact that in overlapping clustering we must take into account the multiplicity of item occurrences in clusters and categories. For instance, if two items share two categories and share just one cluster, then the clustering is not capturing completely the relation between both items (see items 1 and 2 in the second case of Fig. 15). On the other hand, if two items share three clusters but just two categories, then the clustering is introducing more information than necessary. This is the third case in Fig. 15.

These new aspects can be measured in terms of *precision* and *recall* between two items. Let us define:

$$\text{Multiplicity Precision}(e, e') = \frac{\text{Min}(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|C(e) \cap C(e')|}$$

$$\text{Multiplicity Recall}(e, e') = \frac{\text{Min}(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|L(e) \cap L(e')|}$$

where e and e' are two items, $L(e)$ the set of categories and $C(e)$ the set of clusters associated to e. Note that Multiplicity Precision is defined only when e,e' share some cluster, and Multiplicity Recall when e,e' share some category. This is enough to define Bcubed extensions.

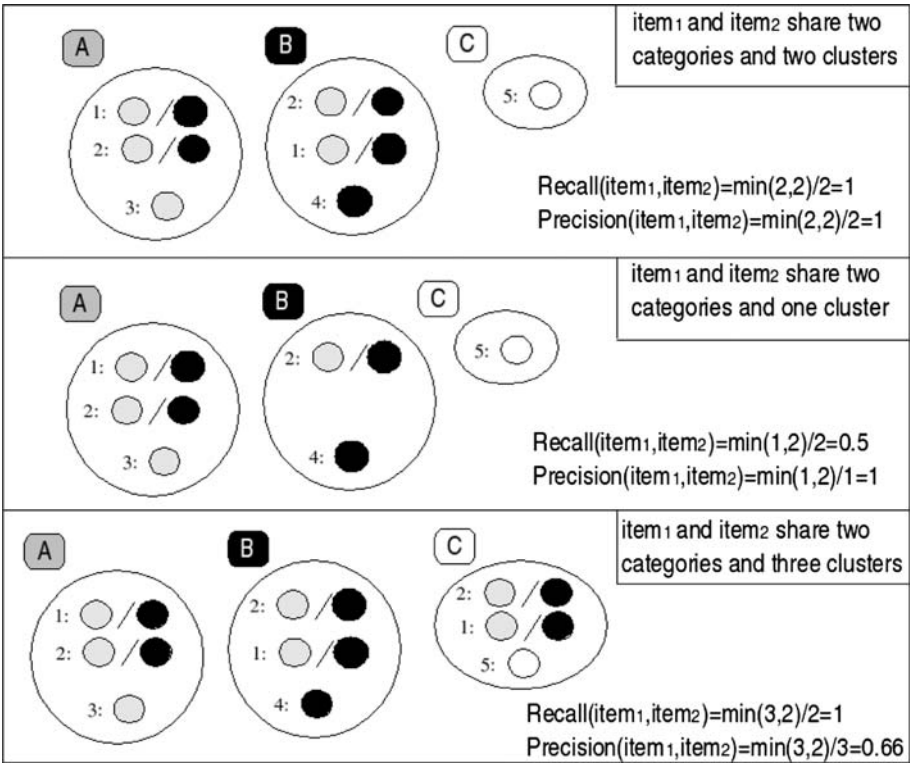


Fig. 15 Computing the multiplicity recall and precision between two items for extended BCubed metrics

Multiplicity Precision is used when two items share one or more clusters, and it is maximal (1) when the number of shared clusters is lower or equal than the number of shared categories, and it is minimal (0) when the two items do not share any category. Reversely, Multiplicity Recall is used when two items share one or more categories, and it is maximal when the the number of shared categories is lower or equal than the number of shared clusters, and it is minimal when the two items do not share any cluster.

Intuitively, multiplicity precision grows if there is a matching category for each cluster where the two items co-occur; multiplicity recall, on the other hand, grows when we add a shared cluster for each category shared by the two items. If we have less shared clusters than needed, we lose recall; if we have less categories than clusters, we lose precision. Figure 15 shows and example on how they are computed.

The next step is integrating multiplicity precision and recall into the overall BCubed metrics. For this, we will use the original Bcubed definitions, but replacing the *Correctness* function with multiplicity precision (for Bcubed precision) and multiplicity Recall (for Bcubed recall). Then, the extended Bcubed precision associated to one item will be its averaged multiplicity precision over other items sharing some of its categories; and the overall *extended Bcubed precision* will be the averaged precision of all items. The *extended BCubed recall* is obtained using the same procedure. Formally:

$$\text{Precision BCubed} = \text{Avg}_e[\text{Avg}_{e'.C(e)\cap C(e')\neq\emptyset}[\text{Multiplicity precision}(e,e')]]$$

$$\text{Recall BCubed} = \text{Avg}_e[\text{Avg}_{e'.L(e)\cap L(e')\neq\emptyset}[\text{Multiplicity recall}(e,e')]]$$

It is important to remember that the metric includes in the computation the relation of each item with itself. That penalizes unnapropriate removal or duplication of a cluster with just one element. Note also that when clusters do not overlap, this extended version of BCubed metrics behaves exactly as the original BCubed metrics do, satisfying all previous constraints.

6.3 Extended BCubed: example of usage

In this section, we will illustrate how BCubed extended metrics behave using an example (see Fig. 16). We start from a correct clustering where seven items are distributed along three clusters. Items 1 and 2 belong at the same time to two categories (black and grey). Since both the categories and the clusters are coherent this distribution has maximum precision and recall.

Now, let us suppose that we duplicate one cluster (black circle in Fig. 17). In this case, the clustering produces more information than the categories require. Therefore, the recall is still maximum, but at the cost of precision. In addition, the more the clusters are duplicated, the more the precision decreases (see Fig. 18). On the other hand, if items belonging to two categories are not duplicated, the clustering provides less information than it should, and BCubed recall decreases (Fig. 19).

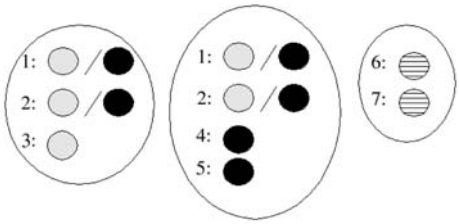


Fig. 16 BCubed computing example 1 (ideal solution): Precision = 1, Recall = 1

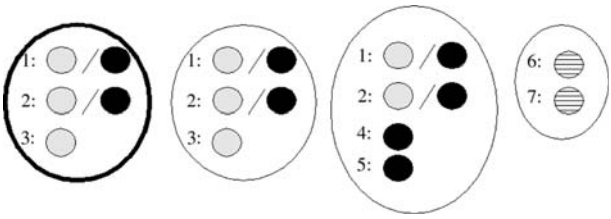


Fig. 17 BCubed computing example 2 (duplicating clusters): Precision = 0.6, Recall = 1

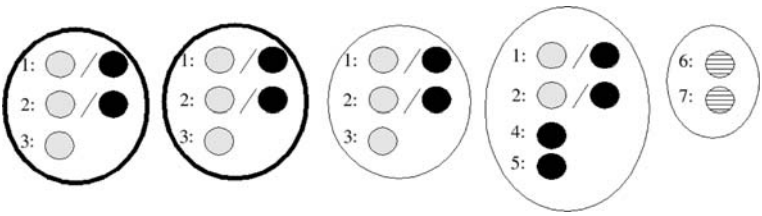


Fig. 18 BCubed computing example 3 (duplicating clusters): Precision = 0.56, Recall = 1

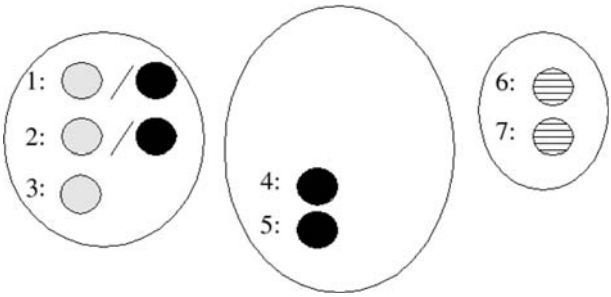


Fig. 19 BCubed computing example 4 (removing item occurrences): Precision = 1, Recall = 0.68

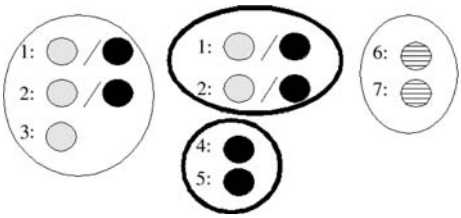


Fig. 20 BCubed computing example 5 (splitting clusters): Precision = 1, Recall = 0.74

If a correct cluster is split, some connections between items are not covered by the clustering distribution and the BCubed recall decreases (Fig. 20). Reversely, if two clusters of the ideal distribution are merged, then some of the new connections will be incorrect, and the multiplicity of some elements will not be covered. Then, both the BCubed precision and recall decreases (Fig. 21).

6.4 Extended Bcubed: a case of study

Here we will compare the behavior of standard metrics Purity and Inverse Purity with the suggested metrics BCubed Precision and Recall, in the context of the analysis of results of

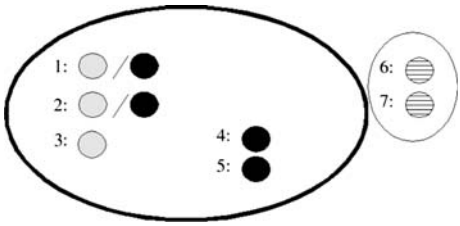


Fig. 21 BCubed computing example 6 (joining clusters): Precision = 0.88, Recall = 0.94

an international competitive evaluation campaign. We exclude from this comparison metrics based on entropy or on counting pairs because they cannot be directly applied to overlapping clustering tasks. We will see that the the standard metrics Purity and Inverse Purity (which were used as official results in the campaign chosen for our study) are not able to discriminate a *cheat* clustering solution from a set of real systems, but the proposed metrics do.

6.4.1 Testbed

Our testbed is the Web People Search (WePS) Task (Artiles et al. 2007) that was held in the framework of the Semeval-2007 Evaluation Workshop.[2] The WePS task consists of disambiguating person names in Web search results. The systems receive as input web pages retrieved by a Web search engine using an ambiguous person name as a query (e.g. "John Smith"). The system output must specify how many different people are referred to by that person name, and assign to each person its corresponding documents. The challenge is to correctly estimate the number of different people (categories) and group documents (items) referring to the same individual. Since the set of different people for each name is not known in advance, there is not a predefined set of categories when grouping items. This can be considered as a clustering task. A special characteristic is that a document can contain mentions to several people sharing the same name (a common example are the URLs with the search results for that name in Amazon). Therefore, this is an overlapping clustering task.

6.4.2 The cheat system

One way of checking the suitability of evaluation metrics consists of introducing undesirable outputs (cheat system) in the evaluation testbed. Our goal is to check which set of metrics is necessary to discriminate these outputs against real systems. Here we will use the cheat system proposed by Paul Kalmar in the context of the evaluation campaign[3] which consists of putting all items into one big cluster, and then duplicating each item in a new, size one cluster (see Fig. 22).

Let us suppose that we are clustering a set of documents retrieved by the query "John Smith". In this case the cheat distribution would imply that every document talks about the same person and, in addition, that every document also talks about another "John Smith" which is only mentioned in that particular document. This solution is very unlikely and, therefore, this cheat system should be ranked in the last positions when compared with real

[2] http://www.nlp.cs.swarthmore.edu/semeval.

[3] Discussion forum of Web People Search Task 2007 (March 23th 2007). http://www.groups.google.com/group/web-people-search-task—semeval-2007/.

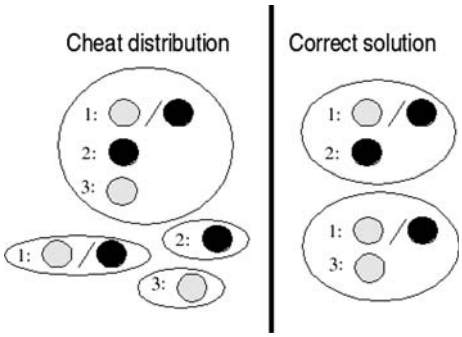


Fig. 22 Output of a cheat system

systems. Purity and Inverse Purity, however, are not able to discriminate this cheat distribution.

6.4.3 Results

Table 2 shows the system rankings according to Purity, Inverse Purity and the F combination of both ($\alpha = 0.5$). The cheat system obtains a maximum Inverse Purity, because all items are connected to each other in the big cluster. On the other hand, all duplicated items in single clusters contribute to the Purity of the global distribution. As a result, the cheat system ranks fifth according to Purity. Finally, it appears in the second position when both metrics are combined with the purity and Inverse Purity F measure.

Let us see the results when using BCubed metrics (Table 3). BCubed Recall behaves similarly to Inverse Purity, ranking the cheat system in first position. BCubed Precision, however, does not behave as Purity. In this case, the cheat system goes down to the end of the ranking. The reason is that BCubed computes the precision of items rather than the

Table 2 WEPS system ranking according to Purity, Inverse Purity and F(Purity, Inverse Purity)

Purity		Inverse Purity		F(Purity, Inverse Purity)	
S4	0.81	Cheat system	1	S1	0.79
S3	0.75	S14	0.95	Cheat system	0.78
S2	0.73	S13	0.93	S3	0.77
S1	0.72	S15	0.91	S2	0.77
Cheat system	0.64	S5	0.9	S4	0.69
S6	0.6	S10	0.89	S5	0.67
S9	0.58	S7	0.88	S6	0.66
S8	0.55	S1	0.88	S7	0.64
S5	0.53	S12	0.83	S8	0.62
S7	0.5	S11	0.82	S9	0.61
S10	0.45	S2	0.82	S10	0.6
S11	0.45	S3	0.8	S11	0.58
S12	0.39	S6	0.73	S12	0.53
S13	0.36	S8	0.71	S13	0.52
S14	0.35	S9	0.64	S14	0.51
S15	0.3	S4	0.6	S15	0.45

Table 3 WEPS system ranking according to Extended BCubed Precision, Extended BCubed Recall, and its F combination

BCubed Precision (BP)		BCubed Recall (BR)		F(Precision, Recall)	
S4	0.79	Cheat system	0.99	S1	0.71
S3	0.68	S14	0.91	S3	0.68
S2	0.68	S13	0.87	S2	0.67
S1	0.67	S15	0.86	S4	0.58
S6	0.59	S5	0.84	S6	0.57
S9	0.53	S10	0.82	S5	0.53
S8	0.5	S1	0.81	S7	0.51
S5	0.43	S7	0.81	S8	0.5
S7	0.42	S12	0.74	S9	0.48
S11	0.36	S11	0.73	S11	0.42
S10	0.29	S2	0.73	S12	0.38
S12	0.29	S3	0.71	S13	0.38
S13	0.28	S6	0.64	S10	0.38
S14	0.26	S8	0.63	S14	0.36
S15	0.23	S9	0.53	S15	0.3
Cheat system	0.17	S4	0.5	Cheat system	0.24

precision of clusters. In the cheat system output, all items are duplicated and inserted into a single cluster, increasing the number of clusters. Therefore, the clustering solution provides more information than required, and the overall BCubed precision of the distribution is dramatically reduced (see Sect. 6.2). On the other hand, the BCubed recall slightly decreases (0.99) because the multiplicity of a few items belonging to more than two categories is not covered by the cheat system.

6.5 Is the problem of overlapping clustering solved?

In this article, we have selected BCubed metrics for extending to overlapping clustering tasks because it satisfies all our proposed constraints for the non overlapping problem, and we have obtained a metric that appears to be more robust than purity and inverse purity.

Note, however, that we should extend and redefine our set of formal constraints in order to know if we have reached a satisfactory solution to the problem. In fact, our metric has at least one problem: a maximal value of Bcubed does not necessarily imply a perfect clustering distribution. This is a basic constraint that is trivially met by all metrics in the non-overlapping problem, but becomes a challenge when overlaps are allowed. Let us illustrate the problem.

In the case of hierarchical clustering, it is easy to show that if extended Bcubed Precision and Recall are maximal (1), then the distribution is perfect. When Precision and Recall are 1, then if two elements share n clusters they must share n categories. Assuming that the distribution has a hierarchical structure, this is equivalent to saying that two elements share a branch of the hierarchy up to level n. One could build up the tree branching, in each step, according to the lenght of the branches shared by each pair of elements to arrive univocally to the ideal clustering. Therefore, maximal BCubed values imply a perfect distribution.

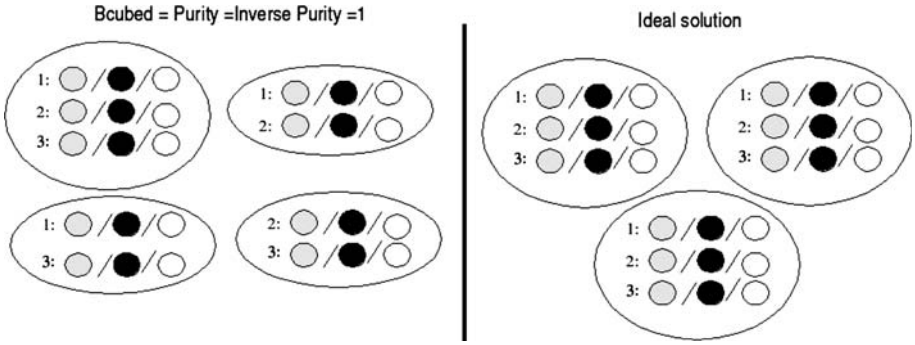


Fig. 23 Counterexample of non-hierarchical overlapping clustering for BCubed and purity-based metrics

Surprisingly, in the case of non-hierarchical clustering none of the current metrics satisfy this basic restriction. Let us illustrate the problem with the example in Fig. 23. All clusters are pure (Purity = 1), and for every category there is a cluster that contains all elements belonging to the category (Inverse Purity = 1). In fact, for every category there is a cluster with maximal precision and recall over elements of that category, and therefore the F-measure (see Sect. 4.1) also achieves a maximal value. In addition, BCubed metrics is also maximal, because the three elements appear three times each, and each pair of elements shares three categories and three clusters. And yet the clustering solution is clearly non-optimal.

Purity and Inverse purity fail to detect the errors because they do not consider multiplicity of occurrences in the elements (see Sect. 6). But BCubed metrics also fail in this case, because they only check coherence between pairs of elements, but this can have crossed relations in different clusters in such a way that they satisfy restrictions on paired elements.

Note, however, that generating such a counterexample requires knowing the ideal distribution beforehand, and therefore this problem cannot lead to a cheat system that gets high scores exploiting this weakness of the metrics. In practice, the possibility of having misleading scores from BCubed metrics is negligible.

7 Conclusions

In this paper, we have analyzed extrinsic clustering evaluation metrics from a formal perspective, proposing a set of constraints that a good evaluation metric should satisfy in a generic clustering problem. Four constraints have been proposed that correspond with basic intuitions about the quality features of a clustering solution, and they have been validated with respect to users' intuitions in a (limited) empirical test. We have also compared our constraints with related work, checking that they cover the basic features proposed in previous related research.

A practical conclusion of our work is that the combination of BCubed precision and recall metrics is the only one that is able to satisfy all constraints (for non-overlapping clustering). We take this result as a recommendation to use BCubed metrics for generic clustering problems. It must be noted, however, that there is a wide range of clustering applications. For certain specific applications, some of the constraints may not apply, and

new constraints may appear, which could make other metrics more suitable in that cases. Some recommendations derived from our study are:

- If the system quality is determined by the most representative cluster for each category, metrics based on matching between clusters and categories can be appropriate (e.g. Purity and Inverse Purity). However, we have to take into account that these metrics do not always detect small improvements in the clustering distribution, and that might have negative implications in the system evaluation/refinement cycles.
- If the system quality is not determined by the most representative cluster for each category, other metric families based on entropy, editing distances, counting pairs, etc. would be more appropriate.
- If the system developer wants to avoid the quadratic effect over cluster sizes (related to our fourth formal constraint), we recommend to avoid using metrics based on counting pairs. Instead of this, the developer may use entropy-based metrics, edit distance metrics or BCubed metrics.
- In addition, if the developer does not want to penalize merging unrelated items in a "rag bag" ("other" or "miscellaneous" cluster), then the only recommendable choice is BCubed metrics.

We have also examined the case of overlapping clustering, where an item can belong to more than one category at once. Most evaluation metrics are not prepared to deal with cluster overlaps and its definition must be extended to handle them (the exception being purity and inverse purity) We have then focused on BCubed metrics, proposing an intuitive extension of BCubed precision and recall that handles overlaps, and that behaves as the original BCubed metrics in the absence of overlapping.

As a case study, we have used the testbed from the WePS competitive evaluation task, where purity and inverse purity (combined via Van Rijsbergen's F) were used for the official system scores. A cheating solution, which receives an unreasonably high F score (rank 2 in the testbed), is detected by the extended Bcubed metrics, which relegate the cheating solution to the last position in the ranking. We have seen, however, that BCubed can, in extreme cases, give maximal values to imperfect clustering solutions. This is an evidence that a complete formal study, similar to the one we have performed for the non-overlapping case, is required.

Three main limitations of our study should be highlighted. The first one is that our formal constraints have been checked against users' intuitions in a limited empirical setting, with just one clustering scenario taken out of a typical ad hoc retrieval test bed, and with a reduced number of users. An extension of our empirical study into different clustering applications should reinforce the validity of our constraints.

The second one is that, beyond formal constraints, there are also other criteria that may apply when selecting a metric. For instance, two important features of any evaluation metric are its ability to scale (v.g. is 0.5 twice as good as 0.25?) and its biases (Strehl 2002). While we believe that our constraints help choosing an adequate metric family, more features must be taken into account to select the individual metric that is best suited for a particular application. In particular, it must be noted that hierarchical clustering, which is a wide information access research area, has peculiarities (in particular regarding the cognitive cost of traversing the hierarchical cluster structures) that need a specific treatment from the point of view of evaluation. Our future work includes the extension of our analysis for hierarchical clustering tasks and metrics.

Finally, note that considering the computational properties of evaluation metrics is beyond the scope of this paper, but might become a limitation for practical purposes.

Indeed, the BCubed metric, which is the best according to our methodology, requires an $O(n^2)$ computation, which is more costly than computing most other metrics (except those based on counting pairs). While the typical amount of manually annotated material is limited, and therefore computing BCubed is not problematic, this might become an issue with, for instance, automatically generated testbeds.

Acknowledgements This work has been partially supported by research grants QEAVIS (TIN2007-67581-C02-01) and INES/Text-Mess (TIN2006-15265-C06-02) from the Spanish government. We are indebted to Fernando López-Ostenero and three anonymous reviewers for their comments on earlier versions of this work, and to Paul Kalmar for suggesting the cheat strategy for the overlapping clustering task.

References

Artiles, J., Gonzalo, J., & Sekine, S. (2007). The Semeval-2007 Weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (Semeval-2007)*, June 23–24 (pp. 64–69). Prague.

Bagga, A., & Baldwin, B. (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 the 17th International Conference on Computational Linguistics (COLING-ACL'98)* (pp. 79–85). Montreal.

Bakus, J., Hussin, M. F., & Kamel, M. (2002). A SOM-based document clustering using phrases. In *Proceedings of the 9th International Conference on Neural Information Procesing (ICONIP'02)* (pp. 2212–2216). Singapore.

Dom, B. (2001). An information-theoretic external cluster-validity measure. IBM Research Report.

Ghosh, J. (2003). Scalable clustering methods for data mining. In N. Ye (Ed.), *Handbook of data mining*. NJ: Lawrence Erlbaum.

Gonzalo, J., & Peters, C. (2005). The impact of evaluation on multilingual text retrieval. In *Proceedings of SIGIR 2005* (pp. 603–604). Salvador de Bahia.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems, 17*(2–3), 107–145.

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Knowledge Discovery and Data Mining* (pp. 16–22). San Diego, CA.

Meila, M. (2003). Comparing clusterings. In *Proceedings of COLT 03*. Washington, DC.

Pantel, P., & Lin, D. (2002). Efficiently clustering documents with committees. In *Proceedings of the PRICAI 2002 7th Pacific Rim International Conference on Artificial Intelligence* (pp. 18–22). Tokyo, Japan.

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (pp. 410–420). Prague.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques, KDD 2000 (pp. 109–110). Boston, MA.

Strehl, A. (2002). *Relationship-based clustering and cluster ensembles for high-dimensional data mining*. PhD thesis, The University of Texas at Austin.

Van Rijsbergen, C. (1974). Foundation of evaluation. *Journal of Documentation, 30*(4), 365–373.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 267–273). NY: ACM Press.

Zhao, Y., & Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Technical Report TR 01-40. Department of Computer Science, University of Minnesota, Minneapolis, MN.