Evaluation of Multiple Clustering Solutions

Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek

Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 München, Germany
http://www.dbs.ifi.lmu.de
{kriegel,schube,zimek}@dbs.ifi.lmu.de

Abstract. Though numerous new clustering algorithms are proposed every year, the fundamental question of the proper way to evaluate new clustering algorithms has not been satisfactorily answered. Common procedures of evaluating a clustering result have several drawbacks. Here, we propose a system that could represent a step forward in addressing open issues (though not resolving all open issues) by bridging the gap between an automatic evaluation using mathematical models or known class labels and the actual human researcher. We introduce an interactive evaluation method where clusters are first rated by the system with respect to their similarity to known results and where "new" results are fed back to the human researcher for inspection. The researcher can then validate and refine these results and re-add them back into the system to improve the evaluation result.

1 Introduction

A major challenge in the development of clustering algorithms is the proper and useful evaluation. In most cases, a clustering algorithm is evaluated using (i) some internal evaluation measure like cohesion, separation, or the silhouette-coefficient (addressing both, cohesion and separation), (ii) some external evaluation measure like accuracy, precision, or recall w.r.t. some given class-structure of the data. In some cases, where evaluation based on class labels does not seem viable, (iii) careful (manual) inspection of clusters shows them to be a somehow meaningful collection of apparently somehow related objects.

All these approaches certainly have their merits but also serious drawbacks.

(i) The evaluation w.r.t. some internal evaluation measure does nothing more than evaluate how well the objective function of the clustering algorithm fits to the chosen evaluation measure. For example, using some compactness measure would be obviously inappropriate to evaluate the results of some density-based clustering [1], simply because density-based clustering does not aim at finding convex clusters. As a consequence, the evaluation does not primarily show that the clustering is meaningful and fitting for the given data. Clusters attributed with good grades could be trivial or rather uninteresting.

(ii) The fundamental problem in using class-labels for evaluation of clustering is the different structure of classes and clusters. Consider for example one

of the best known classification data sets, Fisher's Iris data set [2]. It comprises four descriptors of the Iris flower, namely length and width of petals and sepals, respectively. These descriptors are collected for individual flowers of three different species. The classes are well defined (though not trivial to learn) by some separating borders between members of the classes. The natural clusters in this data set, however, are certainly not evolved according to such (predefined?) borders. Cluster analysis of these data would discover that *I. setosa* is much more different from both, *I. versicolor* and *I. virginica*, than these two are from each other (in fact, they will usually be considered a single cluster). Accordingly, most classification algorithms set out with learning some separating borders between different classes. Opposed to that, clustering algorithms aim at grouping similar objects together. As a consequence, the evaluation of new clustering algorithms towards learning a class structure may introduce some strong bias in the wrong direction into the development and design of new clustering algorithms. Actually, it could be a good and desirable result if a clustering algorithm detects structures considerably different from previously known classes. In that case, the clustering algorithm should not be punished by using some evaluation measure biased towards rediscovery of classes. A more thorough discussion of this issue, along with many more examples, has been provided in [3].

(iii) The third approach, (manual) inspection of clusters and reviewing them for prevalent representation of some meaningful concept, could be figured as 'evaluation by example'. There are attempts to formalize this as 'enrichment' w.r.t. some known concept (this technique is automated to a certain extent in biological analysis of gene data, e.g. [4–10]). In the context of multiple clusterings and overlapping clusters (as are expected in gene data – see the Gene Ontology [11] –, but also in many benchmark data sets that sparked interest of researchers in alternative or multiview clustering, e.g. [12–17], see also [18]) it becomes even more important to find methods of evaluating clusterings w.r.t. each other, w.r.t. existing knowledge, and w.r.t. their usefulness as interpreted by a human researcher. Though the problem of overlapping ground truths (and, hence, the impossibility of using a flat set of class labels directly) is pre-eminent in such research areas as subspace clustering [19], alternative clustering [16], or multiview clustering [13], it is, in our opinion, actually relevant for all non-naïve approaches to clustering that set out to learn something new and interesting about the world (where 'naïve' approaches would require the world to be simple and the truth to be one single flat set of propositions only).

It is our impression, that the third approach is pointing in the right direction since it tries to assess whether some clustering algorithm actually found some new, valid, and previously unknown knowledge (which is, after all, the whole point in performing data mining [20]). As 'evaluation by example', however, it has never been convincingly impartial and always remained tasting somehow subjective and incomplete. The discussion of evaluation scenarios in [3] pointed out some requirements in an automation of evaluation based on multiple (and possibly some unknown) ground truths. Thus we try to establish some first steps in automation of such a process and to set up an evaluation system to address

at least some of the identified requirements. We see this only as some first steps, the system relies on participation of the community to further advance.

In the following, we describe the preliminary system and the envisioned possibilities of future enhancements (Section 2). Based on the available system, we discuss an illustrative example benchmark data set as a case study (Section 3). We conclude the paper in Section 4.

2 A Clustering Evaluation System

Since a main goal of cluster analysis is the discovery of new and previously unknown knowledge, our evaluation concept is built around the comparison of results to known structure in the data. But instead of just computing a score of how well the clustering resembles a known label structure, we actually try to detect situations where it *deviates* from the known structure. Another key difference is that we not only include the target classes, but essentially include any structure information that we can obtain for the data set.

A key source of information are features of any kind. In order to process complex data such as image or video data, feature extraction is essential and a whole research area of its own. But when working in a cluster analysis context, we should treat the features as known properties of the data, and instead evaluate how much *additional* information the clustering algorithm is able to extract from the data that goes beyond the data already extracted using the feature extraction methods: in particular, when feature extraction itself is already very good, almost any clustering algorithm will appear to perform well, but the performance is essentially increased to not more than a naïve statistic on the features.

2.1 Assisted Evaluation

The general process of an assisted evaluation is an iterative interaction between the computer system and the researcher. The system uses the available information to find feature descriptions of the clusters. Clusters that can not be explained sufficiently well using the existing knowledge are then given to the researcher for further external analysis. Knowledge obtained in this process is then added back into the system as additional features, resulting in better explanations for some clusters and thus in new candidates to be analyzed by the human researcher. When assigning a "usefulness" to the different information fed into the system – for example, a simple color feature will not be considered particularly useful but preexisting knowledge – this can also be used to qualitatively rate the output of an algorithm by the usefulness of the information it was able to discover in the data. Both a supervised or semi-supervised evaluation is here possible. For example, the system could perform an initial analysis of the data set, present these results to the analyst, who can then choose results for a more expensive refinement, manually choose complex feature combinations or refine the parameters of found explanations.

2.2 Challenges

In the task of analyzing the characteristics of a cluster, various challenges arise. For example, the cluster size can vary from micro clusters to clusters that span almost the complete data set, resulting in varying imbalance problems. When searching complex explanations involving e.g. the combination of features, or the intersection or union of known classes, the search space is extremely large and an exhaustive search quickly becomes infeasible. Even comparing a cluster with a single numerical feature is non-trivial. Such a feature will give you an ordering (or scoring) of the objects, but the clusters can still occur anywhere within this scoring. Therefore, we need a general way to measure how relevant the information of a feature (or combination of features) is with respect to a particular cluster. Combining such scorings could be based on any linear or non-linear combination of their scores. The scorings however can be strongly correlated, so that in the end, finding the optimal combination offers little benefit to the analyst.

There are many kinds of features, and we will work with two of the most common types of features in the following: class features that differentiate a particular group of objects from the remainder and numerically scoring and ranking features such as the average brightness of an image. Other types such as "bag of words" can probably be handled well enough by breaking them down into individual scoring features.

2.3 Comparing with Existing Classes

Comparing two clusters has of course been extensively studied, and various measures have been developed (see, e.g., [21]). Much of this research (such as pair counting measures) however is designed to compare two complete partitionings of the data (containing more than one cluster each). In our setting, we are again evaluating single clusters with respect to overlapping classes and scorings. Comparing two clusters however is done using simple measures such as precision, recall, or the F-measure (which represents the harmonic mean of these two):

$$F_1 := \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

A nice property of the F-measure is that both trivial solutions (the empty set and the complete data set) score fairly low due to the product in the numerator. Only when both precision and recall are high at the same time, the F-score will be good. When precision equals recall, they will also be equal to the F-score.

We also use this measure in evaluation of a cluster with respect to a scoring, essentially treating these two cases the same, which we will explain next.

2.4 Comparing Clusters with Scorings

A common way of comparing a two-class problem with a scoring is the evaluation using ROC curves. Instead of evaluating a ranking with respect to a class, we

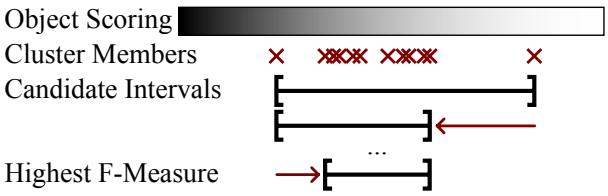


Fig. 1. Evaluating a cluster using the highest F-measure on an interval

could apply ROC curves to evaluate the cluster with respect to the ranking given by the scores. However in our experiments, the results were not very useful: given that the clusters are usually computed on features similar or identical to the reference scorings, a strong correlation and thus a high ROC AUC score between them can be expected. Additionally, ROC is only meaningful when the cluster is at the top or bottom of the scoring, which we would yet have to generalize to allow it to occur at arbitrary positions.

Instead, we chose an approach based on a kind of compactness based on the comparison with classes as discussed before: We search for an arbitrary interval within the scoring that has a high F-measure. Too large an interval will score badly because of a bad precision, while too narrow an interval will suffer from a bad recall. A compact interval containing mostly cluster members however will achieve a high F-measure. In this context, precision can be considered as the density of the cluster members in the interval, while recall is the coverage. Note that this measure is independent of the actual position of the interval within the scoring or the order within the interval. Since we are interested in the *potential* of agreement between the scoring and the cluster, we want to use the maximum F-measure possible; however naïvely there are $O(n^2)$ possible intervals to test. Luckily, we can exploit some monotonicity properties here. Recall obviously is monotonously decreasing, so any subinterval will have at most the same recall. Interesting intervals are thus on the skyline of precision and recall. Intervals which do not have a cluster member on the interval boundary are obviously dominated by the subinterval that fulfills this property (same recall, but better precision). This reduces the search space to $O(k^2)$ for cluster size k. However, we perform a greedy search by starting with the smallest interval containing all cluster members (so at recall 1), then repeatedly narrow down the interval as sketched in Figure 1 by trying to cut off leading and trailing cluster members along with any non-member as long as we can improve the F-measure this way by improving precision at the cost of recall in at most k iterations.

Ties need special handling: an interval may never split within a tie. Then we can map an existing class to a scoring by setting all members to 1 and non-members to 0. If there is some overlap between the test cluster and the known class, the result will be the F-score.

2.5 Scoring Combinations

In addition, we perform a greedy search for a simple additive combination of features. In a preprocessing step, we normalized the scores of each scoring to unit variance to improve results in this step. In the greedy combination phase, we now combine the top matching results by just adding their scores and testing the new scoring. When the combined scoring performs better by a sufficiently large amount, we add it to the candidate list. While we only test a very simple combination of features – not even considering full linear combinations – this greedy search was very successful in our experiments in finding better explanations than single features. We will show examples of this in the next section. But obviously there is much room for improved heuristics in finding such combined explanations.

2.6 Result Presentation

There are essentially infinitely many combinations possible, and even when just using the additive combinations we have theoretically $O(2^r)$ scores for each cluster. The top score itself is often not very useful to the analyst: it may be just one of many very similar explanations. The most interesting analysis results occur when a combination of scorings offers a significantly better explanation than the individual single features, or when there was not found any adequate explanation at all. Therefore we need to make a selection of the results to present to the user. As a heuristic, we will present a result to the user if it is the best single-feature explanation or if no other score with a single feature added or removed performed better. Additionally, we will stop once a threshold of matches has been reached by the accumulated amount explained. Other application-domain specific heuristics may be useful, for example when there is a large number of correlated features.

3 A Case Study

For the case study, we started to analyse the ALOI [22] image data set. It consists of 110250 images of 1000 objects taken from 72 angles and in a series of controlled light conditions varying both color temperature and lighting angle. This metadata can be used to obtain a couple of overlapping classes on the data set, resembling the object number, the viewing angle, the lighting angle, lighting color temperature, and a stereo image shift. Some of these classes are however only useful for machine learning tests; in particular the rotation and stereo image shift usually require a training set and optimized color features.

In addition to these labels we compute some simple color analysis on the pictures. We defined a set of 77 colors spaced evenly in HSV color space (18 hues with 100% and 50% each in saturation and brightness plus 5 grey values for saturation 0%), then computed the average pixel color similarity to these colors for each image to obtain object reference scorings.

For the actual algorithm, we independently produced traditional color histograms in HSV color space with 28 dimensions: 7 bins in hue and 2 bins in saturation and brightness each. In contrast to the features above, the histogram dimensions are not independent, but each pixel is assigned to the closest histogram bin only, so the histograms add up to 1. While the performance of the histograms is of course expected to be similar to the other color features, we wanted to avoid using identical features to not overfit our analysis method.

Early analysis on the objects in this data set allowed us to identify various groups of objects that form sensible clusters aggregating multiple objects such as different jam cans. These additional human-verified clusters sometimes form a hierarchy: for example there are multiple yellow rubber ducks that can be considered a cluster, but there also is a red rubber duck that can be added to form an "any-color rubber duck" cluster. However, there were also some interesting additional features hidden in the data set that were surprisingly useful in explaining results. We highlight these features using a bold typeface and we will explain these features in the discussion below.

We ran OPTICS [23] on the 28 dimensional HSV histograms using Manhattan distance (since this is a rather large data set, and we can use an R^*-tree [24] for acceleration here; on normalized vectors, Manhattan distance equals histogram intersection distance [25]; all implementations featured by ELKI [26]). We chose minPts = 15, $\epsilon = 0.3$ (solely for performance improvements) and $\xi = 0.03$ and obtained a hierarchy of 1442 clusters. The median size is 40 objects, the largest cluster contains 343 images. OPTICS is not a subspace clustering algorithm, but it is a truly hierarchical clustering algorithm, so certain types of overlap among clusters occur. While the majority of objects was not clustered using these unoptimized parameters, the detected clusters were still interesting to analyze. We will give some examples here.

There is a cluster that contains 18 images from object 938 and 19 images from object 939. Some sample images are shown in Figure 2. The cluster is not very surprising, as the two objects are indeed very similar – considering the back side of the objects (images 2(a) and 2(b)). The cluster does not contain the front sides (images 2(c) and 2(d)), which are much more different. In fact, there is another cluster, containing the front sides of object 938 only. The F-measure with the individual clusters is just around 0.25, adding the second object information improves this only slightly to 0.285 (due to the bad recall), but when using both clusters and some color features it rises to above 0.9, offering a much better explanation. Note that one might also be tempted to see a shape cluster, while this is impossible due to the result being computed from color histograms. From the perspective of multiview-clustering, this is a very interesting cluster, since there is a nontrivial cluster that is orthogonal to the original classes, consisting only of parts of the original classes each. As such, the automatic analysis also returns the two matching objects along with color annotations for a best match, thus supporting the analysis as intended. Also note that the reported green colors do not resemble the picture much – but the images may indeed have a very similar distance from this reference color. After our initial analysis we added some new

(a) Object 938 (b) Object 939 (c) Object 938 (d) Object 939

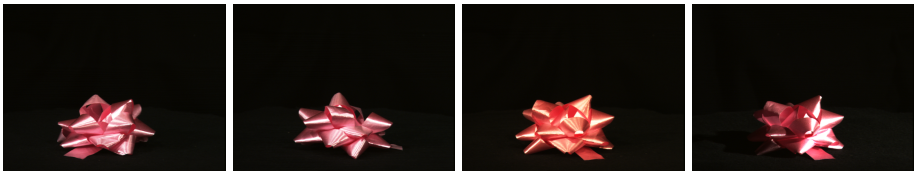
Score	Analysis
0.914	color-408055 color-aaff80 object-938 object-939
0.635	color-808040 color-aaff80 object-938 object-939
0.286	object-938 object-939
0.257	object-939
0.243	object-938
0.877	object-938 object-939 **front-to-back**
0.618	object-938 **front-to-back**
0.590	object-939 **front-to-back**

(e) Analysis result

Fig. 2. Boxes in ALOI image data set

features, including a front-to-back object scoring (ranging from 0 to 1 based on the angle the image was taken from). Including this scoring returns some new explanations. However, they do not score as well as the color-based explanations, making the less interesting color explanation more appropriate. Nevertheless, we already discovered structure in the data that we had not formalized before.

Another cluster (Figure 3) contains 81 images of object 49 and 2 others (so it is almost pure in a traditional sense), but only scores 0.848 on the object itself. Combined with a single color feature, this improves to 0.969. Images 3(a) and 3(b) were both included in the cluster (as were all other rotations and basic color situations). Given the strong uniformity of the object's color representation under rotation, OPTICS cuts off the color variations of the cluster such as image 3(c) (having a light color of $2172K$ as opposed to $3075K$ for the regular images) and angular lighting situations such as image 3(d) (with light coming from the bottom right instead of the center). While the cluster matches the object very well, the actual subset included can be better explained when also using color scorings. Some other objects (e.g. the sea shell 228) were clustered the same way. Furthermore, OPTICS also found a subcluster within this cluster containing just 33 images. The F-measure for the object class on this sub-cluster was just a meagre 0.468. Combining it with a feature that contains only the basic lighting situations, the score rises to 0.606. However, the direct color based explanations match better than the ground truth lighting information, so the clustering algorithm does not appear to have recognized the actual light effects here. In both examples shown so far, the clustering algorithm far from failed: it discovered that there is a *subset* of a class that is more similar to each other than the others.



(a) Regular light (b) Regular light (c) Temp. $2172K$ (d) Light angle $l1c2$

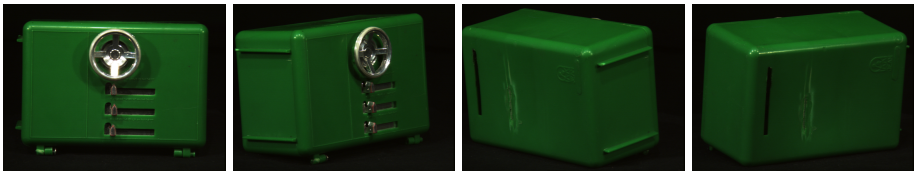
Cluster	Score	Analysis
Parent	0.976	color-2a0080 color-80ff80 object-49
Parent	0.969	color-2a0080 object-49
Parent	0.963	color-aaff00 object-49
Parent	0.848	object-49
Parent	0.936	color-ffd580 object-49 **light-basic**
Parent	0.918	object-49 **light-basic**
Child	0.909	color-008080 color-800080 color-ffd580 object-49
Child	0.899	color-2a0080 color-ffd580 object-49
Child	0.829	color-2a0080 object-49
Child	0.468	object-49
Child	0.925	color-2a0080 color-800080 color-aaff00 object-49 **light-basic**
Child	0.923	color-800080 color-80ff80 object-49 **light-basic**
Child	0.606	object-49 **light-basic**

(e) Analysis of clusters

Fig. 3. Decorative loop in ALOI image data set (Object 49)

In object 492 another interesting hierarchy was discovered (see Figure 4). The outer cluster contains 99 images of the object, while the inner cluster contained just 29. The outer cluster obviously is fairly complete, it just misses some lighting conditions. The inner cluster contains only front views of the object (images 4(a) and 4(b)), but not of the back side (images 4(c) and 4(d)). This is not very surprising, given the silver handle present on the front side of the object, but absent from the back. Note that the inner cluster is explained by colors much better than by the object class, despite being pure, while the outer cluster also scored very well when compared with the object itself. For this cluster again we added the front-to-back object scoring. For the main cluster, this does not improve the result at all (as expected). For the inner cluster, the result however almost doubles, allowing the claim that the algorithm had successfully discovered front views of the object. The result slightly improves with additional color scorings, which is not surprising given that the algorithm had used color information.

Then there is a cluster that caught our attention by having 155 images, making it clearly larger than the expected class size. It contained 74 and 75 images of the objects 981 and 982, respectively, two very similar metal pots (see Figure 5), along with 5 other objects (likely an artifact of the OPTICS ξ "steep up area" definition). The automated analysis suggests that the cluster is based

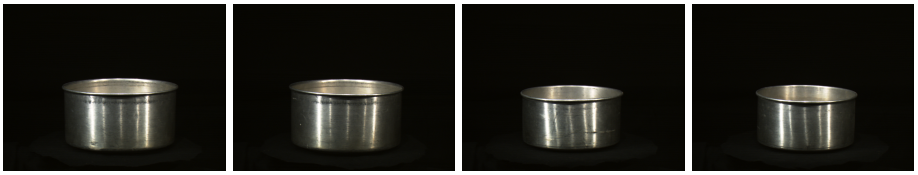


(a) Front of 492 (b) Front of 492 (c) Back of 492 (d) Back of 492

Cluster	Score	Analysis
Parent	0.971	color-00ff00 color-408040 object-492
Parent	0.961	color-00802b color-00ff00 object-492
Parent	0.938	object-492
Parent	0.938	object-492 **front-to-back**
Child	0.812	color-80002b
Child	0.774	color-00802b
Child	0.414	object-492
Child	0.852	color-408040 color-80002b object-492 **front-to-back**
Child	0.846	color-408040 **front-to-back**
Child	0.821	object-492 **front-to-back**

(e) Analysis result

Fig. 4. Green savings box in ALOI image data set



(a) Object 981 (b) Object 981 (c) Object 982 (d) Object 982

Score	Analysis
0.955	color-00802b color-2a8000 object-981 object-982
0.940	color-0000ff object-981 object-982
0.925	color-ffff80 object-981 object-982
0.790	object-981 object-982
0.564	object-982
0.556	object-981
0.974	color-00802b object-981 object-982 **light-basic**
0.939	object-981 object-982 **light-basic**

(e) Analysis result

Fig. 5. Metal pots in ALOI image data set

on the two objects along with color restrictions. However, when adding the basic lighting scoring again, the result is explained better. In retrospection, this is not surprising, given that the metallic object does reflect the light to some extend,

and the object color is thus expected to vary much with the light in contrast to for example the green objects before. Again there is a child cluster and a super cluster which adds 36 images of another metallic object.

4 Conclusion

Building upon some points taken concerning the evaluation of multiple clusterings in last year's MultiClust workshop [3], here we developed some first steps in implementing the vision. We provide a system for evaluation of clusterings, based on prior knowledge as well as on readily extensible knowledge. Currently, the system comprises the ALOI data set. We discussed exemplary clustering results for these data in a case study. The system allows to judge whether some cluster is rather trivial (given it is related to a known concept at all), whether it is a combination of such concepts, or whether it might comprise an interesting, non-trivial, new concept.

During the case study performed on the ALOI data set, we were able to discover nontrivial structure in the data set that we had not been aware of before, but that we were able to formalize and add back into the system to improve the analysis results.

Along with the reference files computed for the ALOI data set, including the advanced structure we found during our analysis, the analysis toolkit is available on the ELKI web page: `http://elki.dbs.ifi.lmu.de/`.

We encourage researchers to use and extend this toolkit for evaluating their findings and for contributing additional structure information. We also would welcome the incorporation of other data sets.

References

1. Kriegel, H.P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. WIREs DMKD **1**(3) (2011) 231–240
2. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7** (1936) 179–188
3. Färber, I., Günnemann, S., Kriegel, H.P., Kröger, P., Müller, E., Schubert, E., Seidl, T., Zimek, A.: On using class-labels in evaluation of clusterings. (2010)
4. Zeeberg, B.R., Feng, W., Wang, G., Wang, M.D., Fojo, A.T., Sunshine, M., Narasimhan, S., Kane, D.W., Reinhold, W.C., Lababidi, S., Bussey, K.J., Riss, J., Barrett, J.C., Weinstein, J.N.: GoMiner: a resource for biological interpretation of genomic and proteomic data. Genome Biology **4**(4:R28) (2003)
5. Al-Shahrour, F., Diaz-Uriarte, R., Dopazo, J.: FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics **20**(4) (2004) 578–580
6. Datta, S., Datta, S.: Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC Bioinformatics **7**(397) (2006)
7. Gat-Viks, I., Sharan, R., Shamir, R.: Scoring clustering solutions by their biological relevance. Bioinformatics **19**(18) (2003) 2381–2389

8. Gibbons, F.D., Roth, F.P.: Judging the quality of gene expression-based clustering methods using gene annotation. Genome Res. **12** (2002) 1574–1581
9. Lee, S.G., Hur, J.U., Kim, Y.S.: A graph-theoretic modeling on GO space for biological interpretation of gene clusters. Bioinformatics **20**(3) (2004) 381–388
10. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Guissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9) (2006) 1122–1129
11. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**(1) (2000) 25–29
12. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proc. ICDM. (2004)
13. Cui, Y., Fern, X.Z., Dy, J.G.: Non-redundant multi-view clustering via orthogonalization. In: Proc. ICDM. (2007)
14. Jain, P., Meka, R., Dhillon, I.S.: Simultaneous unsupervised learning of disparate clusterings. Stat. Anal. Data Min. **1**(3) (2008) 195–210
15. Günnemann, S., Müller, E., Färber, I., Seidl, T.: Detection of orthogonal concepts in subspaces of high dimensional data. In: Proc. CIKM. (2009)
16. Qi, Z.J., Davidson, I.: A principled and flexible framework for finding alternative clusterings. In: Proc. KDD. (2009)
17. Dang, X.H., Bailey, J.: Generation of alternative clusterings using the CAMI approach. In: Proc. SDM. (2010)
18. Kriegel, H.P., Zimek, A.: Subspace clustering, ensemble clustering, alternative clustering, multiview clustering: What can we learn from each other? (2010)
19. Kriegel, H.P., Kröger, P., Zimek, A.: Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM TKDD **3**(1) (2009) 1–58
20. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: Knowledge discovery and data mining: Towards a unifying framework. In: Proc. KDD. (1996)
21. Pfitzner, D., Leibbrandt, R., Powers, D.: Characterization and evaluation of similarity measures for pairs of clusterings. KAIS **19**(3) (2009) 361–394
22. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.: The Amsterdam Library of Object Images. Int. J. Computer Vision **61**(1) (2005) 103–112
23. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering points to identify the clustering structure. In: Proc. SIGMOD. (1999)
24. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-Tree: An efficient and robust access method for points and rectangles. In: Proc. SIGMOD. (1990)
25. Swain, M., Ballard, D.: Color indexing. Int. J. Computer Vision **7**(1) (1991) 11–32
26. Achtert, E., Hettab, A., Kriegel, H.P., Schubert, E., Zimek, A.: Spatial outlier detection: Data, algorithms, visualizations. In: Proc. SSTD. (2011)