Clustering by Intent: A Semi-Supervised Method to Discover Relevant Clusters Incrementally

George Forman[1(✉)], Hila Nachlieli[2], and Renato Keshet[2]

[1] Hewlett-Packard Labs, Palo Alto, USA
george.forman@hp.com
[2] Hewlett-Packard Labs, Haifa, Israel

Abstract. Our business users have often been frustrated with clustering results that do not suit their purpose; when trying to discover clusters of product complaints, the algorithm may return clusters of product models instead. The fundamental issue is that complex text data can be clustered in many different ways, and, really, it is optimistic to expect relevant clusters from an *unsupervised* process, even with parameter tinkering.

We studied this problem in an interactive context and developed an effective solution that re-casts the problem formulation, radically different from traditional or semi-supervised clustering. Given training labels of some known classes, our method incrementally proposes complementary clusters. In tests on various business datasets, we consistently get relevant results and at interactive time scales. This paper describes the method and demonstrates its superior ability using publicly available datasets. For automated evaluation, we devised a unique cluster evaluation framework to match the business user's utility.

Keywords: Semi-supervised clustering · Class discovery · Topic detection

1 Introduction

Hewlett-Packard uses text mining techniques to help analyze customer surveys, customer support logs, engineer repair notes, system logs, etc. [11] Though clustering technologies are employed to discover important topics in the data, usually only a small fraction of the proposed clusters are relevant. This is expected by data mining practitioners, but can prove somewhat disappointing to business users. The fundamental issue is that such complex text data can be clustered in many different ways, and it is unlikely that an *unsupervised* algorithm stumbles upon the one that suits the user's current intent. We have often found they still fail to produce useful clusters even with repeated attempts at adjusting the various parameters by data mining experts.

Furthermore, once some initial large clusters are recognized and dealt with, the remaining data tends to produce decreasingly useful clusters. In fact, sometimes the removal of the known issues causes a shift to less relevant breakdowns

© Springer International Publishing Switzerland 2015
A. Bifet et al. (Eds.): ECML PKDD 2015, Part III, LNAI 9286, pp. 20–36, 2015.
DOI: 10.1007/978-3-319-23461-8_2

of the data, e.g., by setting aside some clusters of known laptop issues (old batteries or cracked displays), the remaining data may be more likely to cluster by product type or geography—frustrating the intent of the user.

One may think that semi-supervised clustering algorithms would provide the answer [2], but they do not. We explored using constrained clustering, a form of semi-supervised learning with `must-link` and `cannot-link` constraints [3,26], but we found its results mostly useless for our purposes (see Tables 1 and 2). Additionally, we considered constrained non-negative matrix factorization (CNMF) methods [8,18]. We tested three implementations, but found both their speed and their results unacceptable. (See the experiments in Section 3.) Fundamentally, most semi-supervised techniques are designed to improve classification, but instead we seek improved discovery of clusters by leveraging the known categories as partial supervision.

Besides the troublingly poor results, we find that clustering solutions tend to be slow.[1] We tried the research software of a half-dozen different publications that claimed to be 'fast'—such as for clustering web search results instantly as they are displayed—but none of them approached the speed needed for interactive use on our text datasets with tens or hundreds of thousands of rows. Research in semi-supervised clustering that involves pairwise constraints typically considers up to thousands of constraints. But once several hundred cases have been labeled for each of a dozen known categories, we end up with millions of pairwise constraints—not very scalable for interactive response times. Also, since clustering into too few clusters will mix different topics together, for our complex data we need to generate many clusters, resulting in linear slowdown for most algorithms. It is a poor interaction: the user waits and waits for the results, then hundreds of clusters appear for the user to examine one by one, the fixed results oblivious to the judgments the user makes as they peruse the voluminous output.

Clustering By Intent (CBI): By examining the practical needs of our interactive users, we reformulated the semi-supervised clustering problem as a substantially transformed data mining task with a distinct yet familiar character, which we shall call *Clustering By Intent*: As the user incrementally explores the dataset, they maintain a growing set of discovered, approved classes, each associated with labeled training cases (typically tens to thousands).[2] The user iteratively requests a cluster, which should be incrementally generated on demand with quick response time. The user may (a) reject the cluster (either being irrelevant or perhaps too impure), (b) accept it as a new class, or (c) merge it into

[1] Witness the large number of clustering publications with *fast*, *efficient*, or *scalable* in their titles, attesting to the problem.

[2] **Terminology:** Let us say that the underlying domain data consists of a set of generally non-overlapping ground-truth *topics* with respect to the user's current intent, e.g., different failure modes, or else product types, or else geographies—not all perspectives mixed together at once. The algorithm strives to return a *cluster* (list of cases) with high *purity*—the precision of the cluster with respect to the cluster's main topic (the most common topic among its cases). The user creates a *class* corresponding to one or more ground-truth topics of interest.

Table 1. Illustrative Comparison of Methods, Clustering By Sport: semi-supervised clustering of 28,166 Reuters news sports headlines, where the supervision given is a single class containing 986 headlines having the word 'baseball.' We show the first 18 outputs of each method, marking (✗) those that are repeats or not relevant. For Constrained K-Means, we report largest clusters first, showing the most distinguishing word of each cluster of documents. For CBI, we limited it to one word per cluster, but the method is more general.

Clustering By Intent	**Constrained K-Means [26]**
soccer	soccer
cricket	cricket
tennis	✗uk
rugby	✗africa
golf	✗first
racing	union (rugby)
skiing	tennis
athletics	racing
basketball	nhl (hockey)
hockey	✗tennis
cycling	✗spain
boxing	✗france
nfl (football)	✗sri (lanka)
swimming	golf
olympics	✗uk
rallying	✗cup
skating	skiing
motorcycling	athletics

an existing class. The algorithm should be responsive to previous user actions, including the most recent supervision.

A couple more points are in order. First, the *purity* of the returned cluster matters greatly to the domain expert. It is easier to recognize a topic if the cluster has high purity, ideally just a single topic. For our typical, complex text domain data, determining the meaning and worth of a proposed cluster can take the user awhile examining its cases. Thus, it is best to provide a manageable list of cases that are most typical or central to the cluster, rather than return a much larger set of cases that may include some other topics mixed in.[3]

Second, the *size* of the cluster topic matters to the user. Although the cluster may be described by a small list of cases, the underlying topic that it informs the user about may be large. We usually encounter complex datasets that have a long-tailed distribution of topic sizes. Users ordinarily prefer to discover the larger topics first, ideally working down the tail in order.[4]

For example, in the application of problem management one wants to discover the most common customer problems in order to address them first or with more

[3] It is useful to provide a symbolic description of the cluster as well, such as which query terms form the cluster or which keywords are most associated.

[4] In some business datasets, we have different priority considerations, but for the scope of this paper, we will use the number of cases in the underlying topic.

Table 2. Clustering the Same Data by Country Instead: Same as Table 1, but here the supervision given to the competing algorithms is a single class containing the 5401 headlines with the word 'UK.'

Clustering By Intent	Constrained K-Means
usa	✗soccer
france	✗division
south (africa)	✗tennis
spain	africa
germany	✗cup
italy	✗tennis
netherlands	✗nhl
zealand	✗union
switzerland	spain
republic (of china)	france
greece	✗standings
portugal	✗baseball
japan	✗cricket
canada	✗cricket
austria	✗golf
australian	✗alpine
indies	✗athletics
belgium	✗basketball

resources. The CBI task fits squarely with this application. Typically once many topics have been discovered, the user would ideally follow it with a period of active learning to expand the training set of the recently defined classes, and finish with a process of machine learning *quantification* [10] to estimate the true size or cost of each class.

The goal of the process is to gain insights from the dataset, and at no stage do we expect to achieve full dataset clustering, as real-world datasets are often not fully clusterable. We do, however, assume that the intent of the user is consistent and does not change viewpoint during the process.

Of course, the user may enact a separate analysis on the same dataset with a different perspective. We illustrate this briefly to show the major benefit of clustering *by intent*. Given a dataset of 28,166 news headlines about sports (from RCV1 [16]), we provide the supervision of a single class of 986 headlines containing the word 'baseball.' With no background knowledge or stopword lists, our CBI method (explained in section 2) iteratively generates the cluster queries shown on the left in Table 1, while the results on the right are generated by the well-known semi-supervised clustering method Constrained K-Means [26] using normalized cosine-similarity as its measure.[5] Alternatively, if the user instead gives the supervision of a single class of the 5401 headlines containing 'UK,' then we get the results in Table 2. The contrast in the CBI outputs for the

[5] We removed stopwords in order to assist Constrained K-Means, at the request of the reviewers; the results are substantially unimproved. (We avoided removing the common stopword 'us' to avoid masking the country 'US'. The ideal algorithm should not need tailored stopword lists in order to find the meaningful terms.)

two intents is night and day, whereas the contrast between the two sets of Constrained K-Means results is weak, and not apparently aligned in any meaningful way to the user's supervision.

The contributions of this paper include: (a) Distinguishing the *Clustering By Intent* data mining task—a new kind of semi-supervised learning. The supervision is given on the known classes and the goal is to discover large unknown topics that are relevant to the user's intent. (b) Detailing how CBI is different from the many recognized data mining tasks—Section 4. (c) Offering a specific CBI algorithm that excels for text domain datasets—Section 2. (d) Illustrating the effectiveness and directability of the method on an intuitive example dataset. (e) Providing a method of automated evaluation for this interactive task without a person in the loop—Section 3.1. (f) Using this method to quantitatively evaluate the algorithm and comparing it with other methods across a gamut of conditions drawn from six publicly available datasets—Section 3. (g) Identifying promising leads for future work—Section 5.

2 CBI Methods

The input to any *Clustering By Intent* method is a typical K-class training set T, plus a set of unlabeled examples U. Not only should one expect that U contains undiscovered classes, but also that some of these unlabeled examples belong to the K known classes. In practical business use, this is particularly the case for periodic analyses where additional unlabeled examples have accumulated for all classes (known and unknown) since the training set was previously developed. Notice that emerging, epidemic topics might have appeared in U since the previous analysis.

The output is an abstract sequence of clusters $C_{0,1,\ldots}$ pulled by the user on demand. A cluster consists of a list of cases of U, and, optionally, a query or description of the topic being proposed. The user may volunteer or implicitly generate feedback on the disposition of any cluster to improve ensuing results. As soon as the training set T is changed, the algorithm retrains.

CBI: We begin by describing one of our leading CBI methods which is appropriate for sparse datasets, such as those that consist of text features in the common *bag of words* representation. (Note that categorical data fields can easily be represented as sparse key=value binary features.) We begin by training a base multi-class classifier that returns the confidence measure for each of its predictions: the margin between the highest scoring class and the runner-up class.[6] We select the low confidence cases of U as our 'residual' set R. The purpose is to avoid cases that are likely to belong to known classes.

Next, when a cluster is demanded, we select cases of the residual R according to the algorithm in Table 3, which also returns a descriptive query. After each cluster is returned, we remove its cases and query terms from R. The

[6] In the rare and short-lived circumstance when only a single class is known, a one-class classifier or a Positive-Unlabeled learner would be appropriate [17].

Table 3. CBI cluster & query construction algorithm

```
1: INPUT: training set T, residual set R from classifier, target cluster size
2: selected cases C = residual set of cases R
3: query = empty list
4: loop
5:    term = highest scoring term wrt C and T (see text)
6:    C' = cases of C containing term
7:    if  |C'| < target cluster size then
8:       RETURN: C, query
9:    end if
10:   C = C'
11:   query += term
12: end loop
```

algorithm iteratively appends terms to a conjunctive query until the resulting set would be below our target cluster size (25). At each iteration, we select the term with highest divergence with respect to C and the training set T. Here we have experimented with a variety of functions, including information gain, chi-squared, bi-normal separation, etc., with some variation. For this paper, we simply use the precision of the term in separating C from false positives matches in T, with a minimum floor of false positives (50).

We have tested various methods for selecting the residual. The experiments section shows results using three separate classifiers: Multinomial Naive Bayes (NB), Regularized Least Squares (RLS), and, as an upper-bound comparison, an oracle classifier (Oracle), which selects all the cases of the unknown topics.

KMeans: For each of these three classifiers independently, we also run the residual through Mini-Batch K-Means (K-Means++ initialization, batch size 400, K=10) [24], returning the clusters largest first. This represents a commonplace workflow: as one recognizes and removes known cases, he or she clusters the remainder to see what else can be found. We also try clustering the entire dataset, which may excel if the data has a propensity to cluster according to the hidden topics.

CNMF: Finally, we tried three different implementations of semi-supervised clustering via Constrained Non-negative Matrix Factorization [8,18]. The experiments show only the best of these.

After illustrating the weak results of Constrained K-Means [3,26] in the introduction and having faced its scalability problems on our larger datasets, we exclude it from the experiments. We have tried a panoply of ideas, but there is space to show only some representatives. Testing other ideas is left as an exercise for the reader. Our implementation leverages the *scikit-learn* package [22].

3 Experiments

There would be no laws and no cricket [without] substantial agreement about what sort of thing cricket ought to be—if, for example, one party thought of it as a species of steeplechase, while another considered it to be something in the nature of a ritual dance... -Dorothy Sayers

Clustering studies ordinarily measure the effectiveness of a method by how well its clusters align with hidden ground-truth class labels in a benchmark dataset, such as by the average purity of its generated clusters. This is philosophically problematic where one dataset may have multiple perspectives of hidden class labels, such as by sports or by countries in our illustration.s[7] Against which standard should an unsupervised clustering output be judged? Given the hidden standard, it makes more sense to grade semi-supervised algorithms, where some ground-truth labels are revealed to impart the desired breakdown.

Existing studies evaluate the set of produced clusters in entirety. What could be wrong with that? It has long been recognized in information retrieval research that it is useful for the objective function to mirror the practical point that a user will need to review the results sequentially. They care much more about the first results than the latter. For this reason, CBI changes the perspective to judging results sequentially. The algorithm must produce a sequence of clusters, not an (unordered) set as traditional methods. Within a single cluster, the user will typically make their judgment about the proposal after reviewing a limited number of cases. Finally, once a topic has been discovered by the user, no credit is awarded for providing additional clusters on the same topic.

Research studies in semi-supervised clustering select training examples at random. Their goal has been to see how much better the clustering results would be if the user would provide just a bit of guidance, preferably applied uniformly. But in our intended use case, our goal is to discover new clusters that are relevant to the current intent. Thus, in CBI the training labels should be drawn from a limited set of classes and credit should not be awarded for returning clusters about the known classes. Furthermore, although we appreciate that obtaining labeled data can be costly in practice, there should be no assumption on the part of the method that the number of labeled cases will be small (it can be easy enough to gather many similar training cases in some domains using a simple binary classifier).

For these reasons and others, we developed a new experiment protocol suited for evaluating CBI tasks. In fact, an important part of the work was to establish an evaluation framework and a credible performance objective measurement in order to then develop and test a wide variety of ideas.

3.1 Experiment Protocol

Our experiment protocol for evaluating CBI methods is shown in Table 4. Note that, since popular topics are easy to discover, we assume the K known classes

[7] For a real, publicly available example, the MULAN dataset repository [25] has a EUR-Lex dataset that has multiple distinct perspectives of labels [19].

Table 4. CBI Experiment Protocol

```
for all benchmark dataset D with each case labeled with a ground-truth topic do
  for # of known classes K = 2, ..., 10 (taking largest topics first) do
    for labeled fraction P = 25%, ..., 100% do
      for all 100 random seeds do
        labeled = randomly select P% of each of the K known classes
        unlabeled = all unselected cases including all unknown dataset topics
        for all method M do
          Train M on (labeled,unlabeled)
          Get the first two cluster outputs of M on the unlabeled data
          Return the better score of the two clusters, scoring each cluster C as:
          score = purity(C.mode)^2 × topic_size(C.mode) / topic_size(largest)
        end for
      end for
    end for
  end for
end for
```

should always correspond to the K largest topics in the dataset, which is often so skewed that one can hardly fail to notice the first couple (see last column of Table 5). Note that the random sampling of labels is a only within each of the K known classes; no labels or class definitions are provided about the remainder of the dataset, not even the number of classes that might be expected. To vary the amount of supervision, one can vary both the number of known classes K as well as the percentage P of each topic's cases that are labeled. In our experiments, we vary one parameter while we hold the other fixed, and vice versa. We use the best score of only the first two clusters output, because we suppose the user is likely to change the training set in some way, and the model would be retrained before producing more outputs.

When it comes to assigning a score to a proposed cluster C, it depends on both size and purity factors. We first determine the cluster's most common represented ground-truth class, C.mode. The *topic size* of the cluster is the number of cases in the ground-truth topic C.mode; note that this is not the size of the cluster C itself. The final score should be directly proportional to the C.mode's topic size, as this is often proportional to real cost. Exceptionally, if C.mode represents a class that is already known in the training set, we give zero credit, in order to align scoring with our purpose of discovering new topics.

The *purity* of a proposed cluster C is evaluated by dividing the number of cases in C.mode by the size of the cluster. We have found that cluster purity matters to the user in a super-linear way: a cluster with, say, 50% purity is less than 50% likely to be understood by the user. Thus, the final score for a cluster

Table 5. Datasets. The last column characterizes the class skew by showing what percent of the dataset falls in the two largest classes.

Dataset	Rows	Features	Classes	K=2
eurlex-codes	16173	5000	20	53%
eurlex-subjects	5418	5000	113	26%
new3	9558	26832	44	13%
fbis	2463	2000	17	36%
re1 (Reuters)	1657	3758	25	42%
wap (web pages)	1560	8460	20	34%

is its purity squared times the topic size of its mode, normalized by the size of the largest unknown topic available to be found (so that finding it achieves 100% credit, rather than have the best possible score shrink as we increase K).

Table 5 shows the six benchmark datasets we use from the text classification domain. The last column characterizes the class skew of each dataset by showing the percent of the rows in the two largest classes. (We verified that the largest classes do not represent a catch-all 'none of the above' class.) The first two datasets are different breakdowns of the EUR-Lex[8] dataset of legal documents: the first by directory codes, the second by subject matter.[9] The remaining four text datasets have been used and described in a variety of other publications (e.g. [10]) and we have provided them in ARFF format at the WEKA dataset repository.[10]

For the first three datasets with >5000 rows, the methods select the residual as the 10% lowest confidence unlabeled cases. But for the three datasets with <2500 rows, 10% returns too few cases to mine. For example, dataset *re1* has 1657 rows and at K=2 already 42% of the dataset is in known classes. So, the true residual is the remaining 961 rows, and that is divided among the 15 remaining classes to be discovered. Selecting just 10% residual at P=75% yields fewer than 150 cases (distributed among all 17 classes)—not enough data. Thus, for the three small datasets, the methods use a threshold of 50%. (Our non-public business datasets usually have tens or hundreds of thousands of rows.)

Figure 1 shows (left) the head of the class distribution for each dataset, and (right) the classifyability of each respective class, as characterized by the F-measure of a NB classifier trying to discriminate that class vs. all others under 4-fold cross-validation. Whereas it is common knowledge that more training examples generally improve classification accuracy for a given class, clearly the difficulty of each individual class can have a larger effect.

[8] http://mulan.sourceforge.net/datasets-mlc.html

[9] In order to fit our experiment harness and reuse classification libraries equipped only for single-label datasets, we simply discarded any rows that actually had multiple labels (thus the number of cases differs).

[10] http://www.cs.waikato.ac.nz/ml/weka/datasets.html

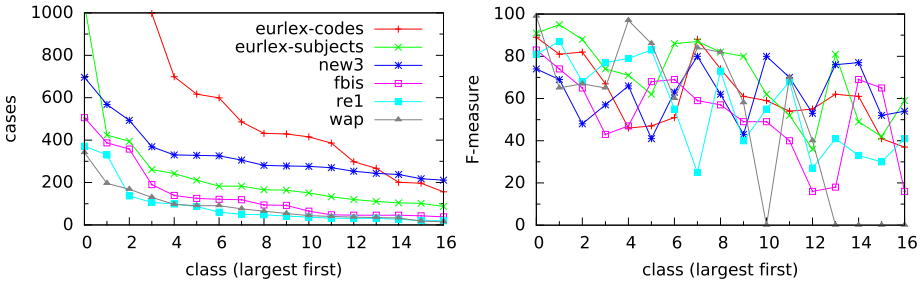

Fig. 1. Dataset characteristics. (left) Class size distribution, shown for the 17 largest classes of each dataset. (right) Classifiability of each class, respectively, as characterized by F-measure of one-vs-all classification by NB classifier under 4-fold cross-validation.

3.2 Results

We begin by comparing the average performance objective (§3.1) of the various methods as we vary the number of known classes K. We hold the percentage of training for each known class at P=75%; for K=2, this results in a supervision level of 10–40% of the dataset. Increasing K provides more training supervision which might benefit the classifier's accuracy, but concomitantly increasing K removes the largest classes from the remaining topics to be found, making the task harder. For different classes, the inherent classification difficulty and clusterability varies, of course. (Refer to Figure 1.) Thus, we expect substantial variation as we change K, not a diagonal trend.

Figure 2 shows the results. We see the semi-supervised method CNMF consistently gave poor results for this task (and this is the best of three implementations). Generally we see CBI methods (black) exceeded KMeans methods (blue), sometimes with the exception of KMeans in combination with the Oracle classifier, which is unachievable in practice. The Oracle classifier (×) did not always lead to the best performance. The classifiers may sometimes do a better job of isolating an interesting and cohesive subset of cases from which to discover a topic. Each of the two classifiers showed many situations where it substantially exceeded the other. In practice one can use cross-validation to select the best model for plain classification tasks, but the lack of training labels for the unknown topics would thwart cross-validation for the CBI objective.

To quantitatively summarize the results across the different datasets and values of K, we computed the average rank of each method, as shown in Figure 3. The red bar indicates those that are not statistically significantly different from the best ranked method, CBI-Oracle, according to the Friedman and Nemenyi tests at alpha=0.05, as prescribed by Demšar [9]. If we were to exclude the two Oracle methods for being impossible in practice, the two CBI methods are better than statistically significantly better than all other methods by a statistically significant margin.

Next we vary the percentage P of training labels provided for each known class, while we fix the number of known classes at K=4. Figure 4 shows these results. They have less variation than the previous results, because the known

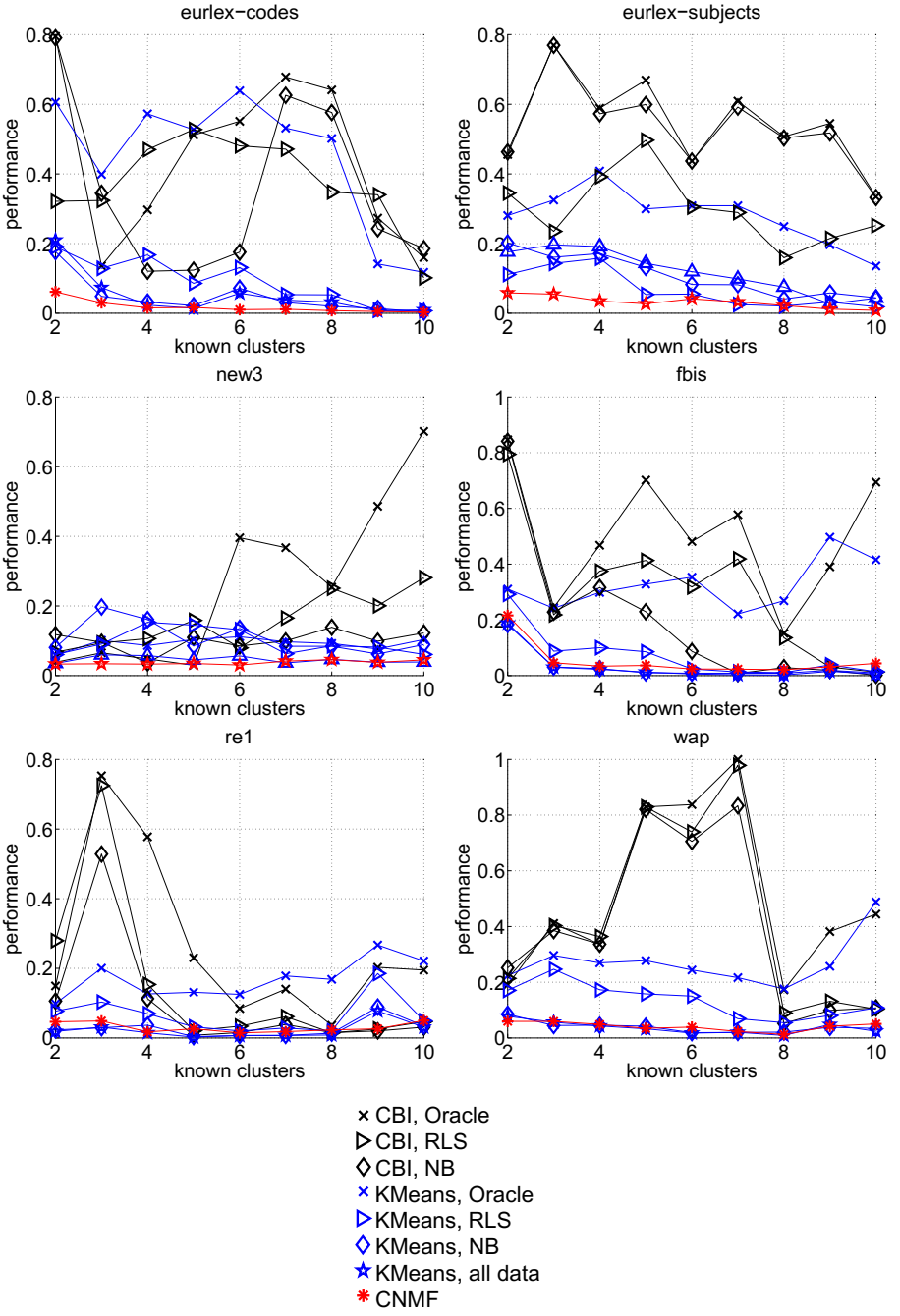

Fig. 2. Performance of all methods on each dataset as we vary the number of known classes K, fixing P=75% for training.

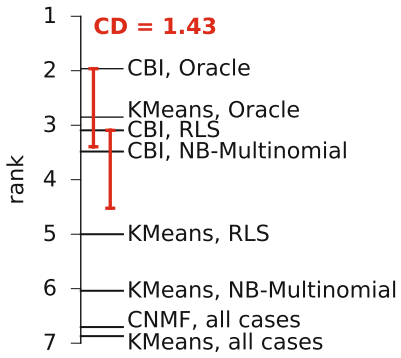

Fig. 3. Average rank and statistically significance differences. The red bar indicates that CBI-RLS is not statistically significantly worse than the oracle, and is significantly better than the state-of-the-art semi-supervised methods.

classes and unknown topics to be discovered are stable across the x-axis in each graph.

In contrast to learning curve studies in classification, it is most striking that increased supervision did not consistently lead to better performance for the CBI objective, though it often helped somewhat (even for CNMF). Labeling more cases removes them from the unlabeled set, reducing the risk for all methods of accidentally proposing a class that is already known. Separately we validated that, as the training level increases, the classifiers showed increasing precision in selecting a residual subset.[11]

Thus, clearly classifier accuracy is not the overriding priority for this task. Case in point, the Oracle returns a perfect residual regardless of training set, yet this does not ultimately lead to the best performance; the size and makeup of the training set affects the CBI method substantially and non-linearly. Increasing supervision benefits CBI-NB for datasets eurlex-subjects, fbis, re1, and wap.

Some of our CNMF results took 5–20 minutes to compute, and RLS classifier training for K=10 took minutes sometimes. This paper focuses on introducing the CBI task and on satisfying the performance objective; not on speed. That said, the CBI process ultimately needs to be put into an interactive loop in applications. For the most part we have not concerned ourselves with fast implementations, but we have prototyped a fast version of a text-based CBI method which clustered 40,000 rows of text data in under 100 ms—clearly suitable for interactive time scales. The CBI algorithm is linear in the number of documents and the number of frequent terms.

[11] Note that this is a different objective than their classification accuracy on the known classes, which is of little interest to us in this paper.

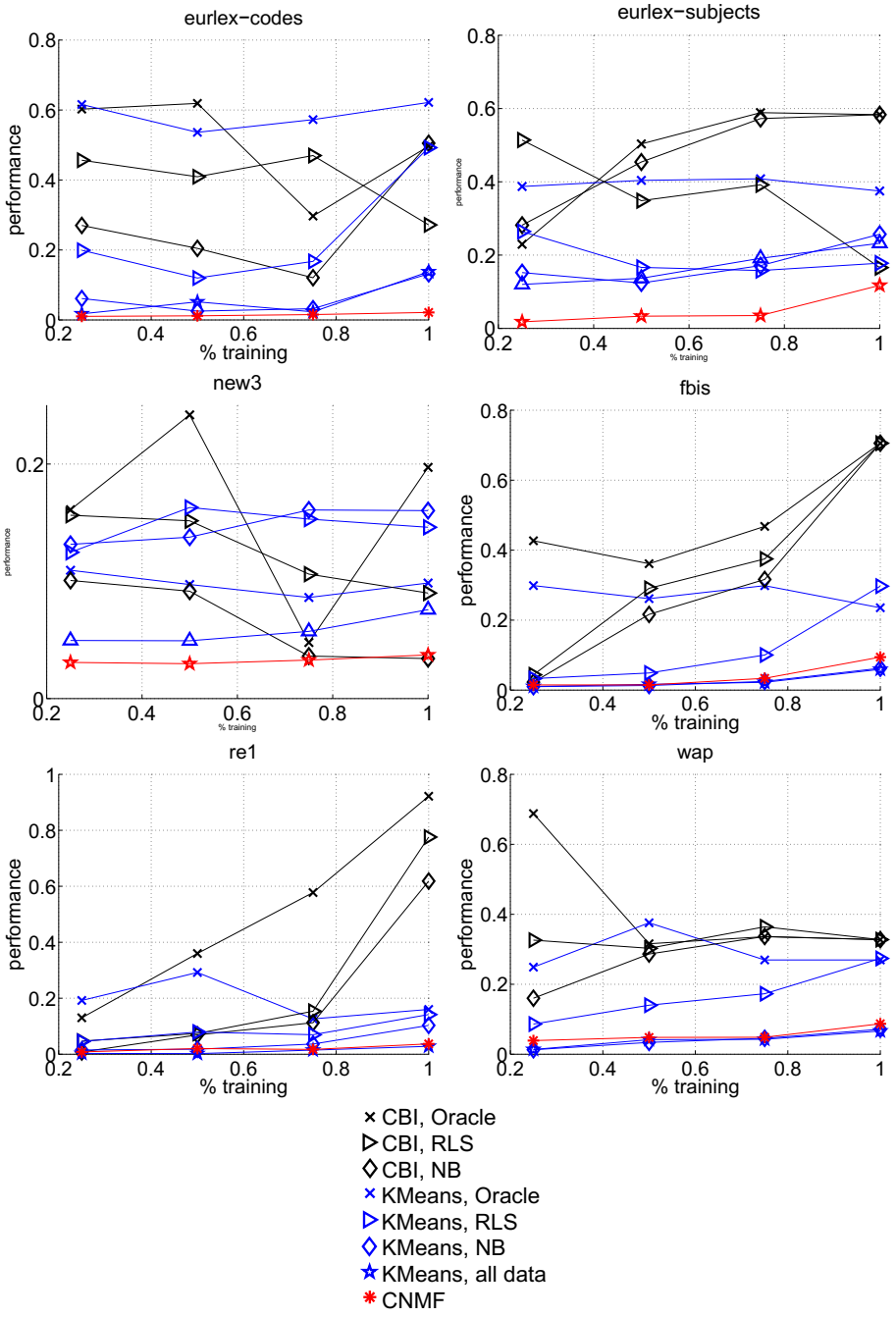

Fig. 4. Performance of all methods on each dataset as we vary the percentage P of training examples randomly selected from the K known classes, fixing K=4.

4 Related Work

The field of *topic detection* has similar goals to CBI: automatically finding new topically related material in streams of data [1]. The streams may be news feeds or Twitter streams [6]. Given their temporal impermanence, the clustering is expected to succeed without supervision, as it does well for Google News. In these domains, news articles are often copied verbatim and so they cluster very neatly. By contrast, CBI targets complex data that cannot be decomposed into user-relevant clusters without some guidance about the user's intent.

Semi-Supervised Learning (SSL) is a broad area, most of which augments labeled training sets with unlabeled data in order to improve the accuracy of classification [7,27]. The minority of research that focuses on semi-supervised *clustering* is described only somewhat differently: augmenting the unlabeled data with some few labels or, more typically, `must-link` and `cannot-link` pairwise constraints in order to improve the clustering. Examples include Constrained K-Means [3,26] and Constrained Non-negative Matrix Factorization [8,18]. Even so, researchers measure the clustering quality by its conformance to the (mostly) hidden ground-truth labels, e.g. classification accuracy, so it ends up being much like the SSL research for classification (e.g. [8]). Furthermore, the accepted experiment methodology uses random sampling to select the supervision. This leads to a high likelihood of covering most large classes, which have the greatest effect on the performance measurement. Thus, supervision is given for all the clusters that should be found, but in CBI we only have supervision for the clusters we already know about and not on the unknown clusters that need to be discovered.

Thus, none of this work is like our CBI framework, which might be said to use extremely 'skewed' supervision and does not hinge upon overall classification accuracy. Under CBI, the clustering algorithm gets no credit for returning clusters on classes that have training examples—this is the domain of classifiers (which can probably perform the job better). As we have seen, if the supervision is focused on only a few known classes, it does not seem to help CNMF or Constrained K-Means to adopt the perspective of the user's intent.

One-class classification, *outlier detection*, *anomaly detection* and *novelty detection* aim to recognize abnormal data points, generally with the assumption that they are rare events and not available in quantity at training time [23]. These methods are considered successful by recognizing individual test cases that are highly unusual; they do not attempt to cluster such cases into meaningful subgroups, as CBI.

Similarly, there are a variety of works in *novel class discovery* that attempt to seek individual cases that may stem from unknown classes [4,13,20]. These methods generally interact with a user via active learning and assume that the user can recognize a novel class when presented with an individual anomalous case. Thus, they evaluate their work as a learning curve showing the number of novel classes encountered over time. By contrast, in CBI there are generally many unknown classes, mostly very small ones in the long tail, and the goal is to find the larger ones. Further, we have the need to present a collection of similar cases to communicate the topic to the user.

The area of *subgroup discovery* sounds entirely appropriate to our goals, but it is actually an unsupervised task that attempts to find rules of interest associated with a feature of a dataset [14,15]. Gamberger et al. even use the title 'expert-guided subgroup discovery', but by this they mean 'the decision of which subgroups will be selected to form the final solution is left to the expert' [12]. *Contrast set mining* and *emerging pattern mining* are variants that seek rules that find significant differences in databases, such as old and new datasets [21]. None of these methods take in the multi-class supervised data of CBI, although perhaps it would be interesting future work to see whether they could be adapted to produce useful results for CBI tasks.

Finally, there is the idea of *meta-clustering*, which takes many different unsupervised clustering results and produces clusters of them which the user can select among [5]. Ideally one could imagine that different meta-clusters would correspond to different user intents. We have not tried it, but without supervision it seems unlikely to produce results nearly as relevant as CBI, even if the user could determine the most appropriate meta-cluster.

As we mentioned earlier, the area of *PU Learning*—learning from positive labeled cases and unlabeled cases, some of which may actually be positives—may be pertinent in the classification stage of CBI at first when the user has only identified a single class. PU Learning addresses binary classification problems, and sometimes considered streams that may have concept drift over time [17], which is ultimately also an issue of concern for the real-world business user.

5 Conclusions and Future Work

We have labored—and made our internal business users to labor—under poor clustering results for years when seeking to discover new clusters relevant to a particular purpose. We found semi-supervised clustering methods intellectually promising, but, unfortunately, we saw little benefit in practice. By stepping back from our assumptions and recasting the task substantially, we have been able to crack a variety of business datasets, and with a natural user-interaction that is quick. This paper elucidates the task, a suitable method, and its evaluation with a novel protocol devised to work easily with any publicly available multi-class benchmark dataset.

We expect future work in this area to compare additional methods, improve on these results, relax assumptions, and remove limitations. In particular, though our focus has been on the text domain, this could be broadened. In order to perform model selection and tuning in practice, future work could develop a form of cross-validation for CBI tasks—a challenge, having no labels available for the long tail. Next, although in this paper the value of discovering a topic was posited to be proportional to its size, in some datasets, we have a cost indicator associated with each case—such as parts and labor costs to resolve each case. In these settings, the total cost of the topic is a more appropriate indicator of value than simply the number of cases in the topic. Cost quantification techniques [10] could also be applied to prioritize probabilistic cluster definitions. Finally, future work

may add emerging topics and/or concept drift to the evaluation with methods to handle them.

Lastly, a philosophical remark. While classification excels at the 'more like this' task, and clustering could be said to excel at the 'find various topics' task, CBI provides a qualitatively new capability: 'find topics different from those, yet alike in some important way.' Even so, it has no higher level concept of how those things are alike. For example, in the sports illustration of Table 1, *we* know these are sports, but there is no runtime representation of this meta-information.

References

1. Allan, J. (eds.): Topic Detection and Tracking, The Information Retrieval Series, vol. 12 Springer (2002)
2. Bair, E.: Semi-supervised clustering methods. Wiley Interdisciplinary Reviews: Computational Statistics **5**(5), 349–361 (2013)
3. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: KDD 2004, pp. 59–68 (2004)
4. Bouveyron, C.: Adaptive mixture discriminant analysis for supervised learning with unobserved classes. J. Classif. **31**(1), 49–84 (2014)
5. Caruana, R., Elhawary, M., Nguyen, N., Smith, C.: Meta clustering. In: ICDM 2006, pp. 107–118 (2006)
6. Cataldi, M., Di Caro, L., Schifanella, C.: Emerging topic detection on twitter based on temporaland social terms. In: MDMKDD 2010, pp. 4:1–4:10 (2010)
7. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised Learning. Adaptive computation and machine learning. MIT Press (2006)
8. Chen, Y., Rege, M., Dong, M., Hua, J.: Non-negative matrix factorization for semi-supervised data clustering. KAIS **17**, 355–379 (2008)
9. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. JMLR **7**, 1–30 (2006)
10. Forman, G.: Quantifying trends accurately despite classifier error and class imbalance. In: KDD 2006, pp. 157–166 (2006)
11. Forman, G., Kirshenbaum, E., Suermondt, J.: Pragmatic text mining: minimizing human effort to quantify many issues in call logs. In: KDD 2006, pp. 852–861 (2006)
12. Gamberger, D., Lavrac, N.: Expert-guided subgroup discovery: Methodology and application. J. AI Research **17**(1), 501–527 (2002)
13. Haines, T.S., Xiang, T.: Active rare class discovery and classification using dirichlet processes. Int. J. Computer Vision **106**(3), 315–331 (2014)
14. Herrera, F., et al.: An overview on subgroup discovery: Foundations and applications. Knowledge and Information Systems **29**(3), 495–525 (2011)
15. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. JMLR **5**, 153–188 (2004)
16. Lewis, D., et al.: RCV1: A new benchmark collection for text categorization research. JMLR **5**, 361–397 (2004)
17. Li, X., Yu, P.S., Liu, B., Ng, S.: Positive unlabeled learning for data stream classification. In: SIAM 2009, pp. 259–270 (2009)
18. Liu, H., Wu, Z.: Non-negative matrix factorization with constraints. In: AAAI 2010, pp. 506–511 (2010)

19. Mencía, E.L., Fürnkranz, J.: Efficient pairwise multilabel classification for large-scale problems in the legal domain. In: ECML/PKDD 2008, pp. 50–65 (2008)
20. Miller, D.J., Browning, J.: A mixture model and em-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets. IEEE Trans. Pattern Anal. Mach. Intell. **25**(11), 1468–1483 (2003)
21. Novak, P.K., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. JMLR **10**, 377–403 (2009)
22. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. JMLR **12**, 2825–2830 (2011)
23. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. Signal Processing **99**, 215–249 (2014)
24. Sculley, D.: Web-scale K-means clustering. In: WWW 2010, pp. 1177–1178 (2010)
25. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. Data Mining and Knowledge Discovery Handbook
26. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-means clustering with background knowledge. In: ICML 2001, pp. 577–584 (2001)
27. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005)