Avoiding Bias in Text Clustering Using Constrained K-means and May-Not-Links

M. Eduardo Ares, Javier Parapar, and Álvaro Barreiro

IRLab, Department of Computer Science, University of A Coruña, Spain
{maresb,javierparapar,barreiro}@udc.es

Abstract. In this paper we present a new clustering algorithm which extends the traditional batch k-means enabling the introduction of domain knowledge in the form of Must, Cannot, May and May-Not rules between the data points. Besides, we have applied the presented method to the task of avoiding bias in clustering. Evaluation carried out in standard collections showed considerable improvements in effectiveness against previous constrained and non-constrained algorithms for the given task.

1 Introduction

Clustering [1] and classification [2] methods have been demonstrated as useful tools in several fields within computer science like Data-Mining (DM) or Information Retrieval (IR). The need for methods for automatic data analysis has arisen when working with large collections of heterogeneous data, where doing it manually by experts was unfeasible. Even though the main difference between clustering and classification has been that the later is performed without any prior knowledge of the data, adding some domain knowledge to the clustering algorithms can result in an considerable effectiveness improvement. This is the idea behind a new family of methods coined as *constrained clustering* [3], where the domain knowledge is introduced as rules in a generalised framework keeping the algorithm domain-independent. Two clear examples of this situation could be clustering data from multiple evidences of information, where introducing guiding data can be very useful, or in collections where the data has a very obvious grouping to which the traditional algorithm are biased, and where more interesting results could be found if we tell the algorithm to avoid that clustering.

These methods, called *"semi-supervised clustering"*, use background knowledge to impose some restrictions on the process, trying to influence the grouping that it finds in the data. This has been a very fruitful field in the last years [4–12]. This constrained clustering is quite different from a classification process, as the domain knowledge gives the clustering algorithm rules over data instances (documents), instead of examples of the categories. These rules reflect some preferences about whether or not the data instances should be in the same cluster, but it is still the algorithm which finds the groups in the data.

In this paper we propose a new framework of constrained clustering which, based on batch k-means, incorporates May and May-Not Link constraints as well as the Must and Cannot Link constraints proposed by Wagstaff et al. in [7], because in most real cases the domain knowledge is not categorical and only hints some traces or patterns. Thus, using absolute constraints could harm the algorithm effectiveness. Another contribution of this work is including unidirectional constraints, which could be interesting when working in certain domains.

After defining the new approach we tested it in an avoiding bias problem. In this real world clustering problem, the traditional algorithms tend to be biased to a dominant grouping, which is also well known, and the objective is to avoid that one, to discover new data interpretations. Our results in this experiment outperformed the Conditional Information Bottleneck-based method (CIB) [9], used as baseline. We also tested in other experiment the behaviour of the algorithm as the number of negative absolute and soft constraints is increased.

Next, in section 2 is presented the new framework. Section 3 describes the experiments and comments the results. Section 4 is devoted to the previous work about semi-supervised clustering, showing the differences with the proposed method. Finally, conclusions are reported in Sections 5.

2 K-means with Absolute and Soft Constraints

The k-means [13] algorithm is a very popular clustering method, due to its good trade-off between effectiveness and cost. It is a generic algorithm, which does not need any prior knowledge apart from the desired number of clusters. Moreover, its clear structure and flow makes extending and modifying it very easy.

In [7] Wagstaff et al. introduced in batch k-means two kinds of bidirectional instance level pairwise constraints, which were previously presented in [6]: *Must-Links*, connecting documents which must be in the same cluster and *Cannot-Links*, connecting documents which must not be in the same cluster. These constraints are absolute, i.e. a clustering has to fulfil all of them to be acceptable. While this absoluteness can be very convenient if we know categorically the relations between instances and we can not afford to have them misplaced, it could represent an excessive burden to the process. Indeed, as the authors admit in [7], it can lead to situations where, even though there is an acceptable solution, it can not be found as the outcome of the algorithm is extremely sensitive to the order in which the documents are inspected. For instance, it could be impossible to find a cluster for a document due to having a Cannot-Link constraint with a document in each cluster, a situation that might have not arisen if we had inspected the "conflictive" document earlier. Even when a solution can be found, the combination of absoluteness and sensitiveness to order can make the presence of constraints more detrimental than beneficial. For example, data instances connected with Must-Links will be dragged unconditionally to the cluster where the first of them is assigned, which could lead to worse clusterings.

In order to overcome these limitations we introduce in this paper two new kinds of soft (non-absolute) constraints, which will influence gradually the process instead of defining categorically where a document must or must not go: *May-links*, connecting documents a and b if a is likely to be in the same cluster as b, and *May-Not-Links*, connecting documents a and b if a is not likely to be in the same cluster as b. These constraints are unidirectional, i.e, we are dealing with ordered pairs. In most domains the constraints will be reciprocal that is, (a, b) and (b, a) would be present. However, there could be others where this capability to express non-reciprocal constraints could be interesting. For instance, consider we want to cluster companies web-pages by industrial sector. It is sensible to assume that the pages of a company's products should be in the same cluster as their company main-page but not the opposite. This knowledge can be represented by a set of May-Links $(product_i, company_x)$. Another difference with the absolute constraints is that the May-Link and May-Not-Link constraints do not necessarily define a transitive relation.

```
CLUSTER($\{x_1, \ldots, x_n\}, k, musts, cannots, mays, mayNots, w$)
  $new \leftarrow$ SELECTRANDOMSEEDS($\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, k$)
  while convergence criterion has not been met
  do $current \leftarrow new$
     $old \leftarrow new$
     CLEAR($new$)
     for $i \leftarrow 1$ to $n$
     do
        $assigned \leftarrow$ ASSIGN($x_i, k, new, current, old, musts, cannots, mays, mayNots, w$)
        if $not(assigned)$
          then error "Impossible to cluster"
     end
  end
  return $new$

ASSIGN($x, k, new, current, old, musts, cannots, mays, mayNots, w$)
  $scores \leftarrow [0, 0, ..., 0]$
  $assigned \leftarrow false$
  for $j \leftarrow 1$ to $k$
  do
     if $\exists x_i \in new[j]$ such that $(x, x_i) \in musts$
       then PUT($x, max, new, current, old$); return $true$
     if $\exists x_i \in new[j]$ such that $(x, x_i) \in cannots$
       then continue
     $assigned \leftarrow true$
     $scores[j] \leftarrow$ SIMILARITY($x$, CENTROID($old[j]$))
     for $h \leftarrow 1$ to $|current[j]|$
     do
        if $\exists(x, current[j][h]) \in mays$
          then $scores[j] \leftarrow scores[j] + w$
        if $\exists(x, current[j][h]) \in mayNots$
          then $scores[j] \leftarrow scores[j] - w$
     end
  end
  if $assigned$
    then $max = indexof(\max(scores))$
         PUT($x, max, new, current, old$); return $true$
    else return $false$

PUT($x, i, new, current, old$)
  $current[clusterof(x, old)] \leftarrow current[clusterof(x, old)] \setminus \{x\}$
  $current[i] \leftarrow current[i] \cup \{x\}$
  $new[i] \leftarrow new[i] \cup \{x\}$
```

Fig. 1. k-means clustering algorithm with Must, Cannot, May and May-Not Links

The New Constrained k-means Algorithm. The resulting algorithm after introducing the absolute and soft constraints in the schema of the batch k-means is detailed in Fig. 1, extending the implementation of constrained k-means introduced in [7]. The input data and parameters are: $\{x_1, \ldots, x_n\}$, the set of documents in the collection to cluster; k, the number of clusters that the algorithm will try to find; $musts$, $cannots$, $mays$ and $mayNots$, the background knowledge in form of constraints to be taken into account and w, the factor of influence of the soft constraints. The constraints $musts$, $cannots$, $mays$ and $mayNots$ are represented as sets of ordered pairs (in $musts$ and $cannots$ we will assume that a previous transitive closure has been taken and that, due their reciprocity, if (a, b) appears, (b, a) appears as well).

The first step (1) is initialising each cluster with a different document chosen randomly from the set of documents to cluster, as a sort of "iteration -1". Afterwards, and until the algorithm satisfies the convergence criterion (2) a loop is executed, where in each iteration the documents are assigned to a cluster using the function Assign, using the outcome of the previous iteration (old), the location of the documents already assigned in this iteration (new) and the previous set of clusters actualised by the changes made in this iteration ($current$).

Given a document x, the function Assign determines to which cluster it should be assigned. For each cluster j (3), the function tries first to honour the absolute constraints that affect x as in Wagstaff et al. [7] . That is, if x has a Must-Link with any of the documents already assigned to cluster j in this iteration (5), x is Put in that cluster and the function returns (6). Also, if there is a document with which x has a Cannot-Link (7), the cluster is discarded.

After testing the absolute constraints, the similarity of x with the centroid of the old cluster is calculated (10). This similarity value ($scores[j]$) will be modified by the soft constraints (13-16) affecting x. For each document which has been already assigned to this cluster in this iteration or has not yet been inspected and with which x has a May-Link, the *score* of the cluster j is increased in a certain amount w. If it has a May-Not-Link, the *score* of the cluster j is decreased in a certain amount w. This strategy fits well with the mechanism of the k-means algorithm, which uses information from an iteration (the centroid of the documents) in order to rearrange them in the next. Moreover, along with the non-absoluteness of the constraints, it lets the sole presence of these constraints affect gradually the clustering process while avoiding the problems exposed earlier. Once those steps have been tried on each cluster, the one with the highest *score* is chosen as the destination of x (20,21). If all the clusters were discarded the appropriate flag is returned (22), aborting the execution of the algorithm.

This new algorithm maintains the good computational behaviour of batch k-means: considering that k is the desired number of clusters, i the number of iterations, c is the number of constraints, and n the number of documents in the collection, our constrained k-means still is $O(k \times i \times n)$ in time. The searches in the constraint lists are not considered because compared with the document similarity calculation their cost is negligible. The algorithm is $O(k + n + c)$ in space, although again can be considered $O(k + n)$ because the space of storing the constrains is much smaller than the space for the documents.

3 Experiments and Results

The clustering algorithms try to detect an underlying organisation in the given data. Often, there is an obvious grouping of it, which is easily found by a simple manual examination. In that case, the clustering algorithms will be probably biased to fall in that organisation, which is not very helpful. The task of avoiding this grouping, trying to make the algorithm pay attention to other facts which could lead it to another unknown clustering, is called "Avoiding Bias", which, as well as having its intrinsic interest, will be used here to show the effectiveness of our constrained clustering algorithm. Besides, we also contribute a comparison of the behaviour of the Cannot and May-Not Links in a similar way as in [7].

In our experiments we have used two datasets used by Gondek and Hofmann in [9]: the first one (i) was created from WebKB's Universities dataset, taking only the documents from Cornell, Texas, Washington and Wisconsin universities and dropping those corresponding to "misc", "other" and "department" (1087 documents). The second one (ii) was created from Reuters RCV1 dataset, taking the documents with only one topic and region label and whose topic is MCAT or GCAT and whose region is UK or INDIA (1600 documents). As in [14], we have used as document representation the Mutual Information (MI) between a document and its terms. Cosine distance was used as similarity measure.

To compare the clustering yielded by the algorithm with a certain reference we have used three metrics [15], where higher values mean more similarity: Purity (P), a precision metric which measures how well the clustering results match the manual split in average, Mutual Information (MI), a metric which measures how much information about a clustering is conveyed by another and Rand Index (RI), which measures the ratio of good decisions made by the algorithm.

Experiment 1: Avoiding Bias. In this experiment we have used the datasets defined above in order to address an Avoiding Bias problem. Each document is categorised according to two different criteria, so we will take one of these criteria as the known clustering of the data and we will try to avoid it, using the constrained k-means algorithm that we have introduced. After the algorithm is executed, we will measure the similarity of the final set of clusters with the known clustering and with the other one present in the data.

The constraints set is created with two May-Not-Link constraints for each pair of documents (i.e. both directions) belonging to the same cluster in the clustering we are trying to avoid (which is already known for us). These are the only constraints that are going to be used in the clustering process. Specifically, the Cannot-Link constraints are unsuitable for this task due to their absoluteness.

In order to produce a fair comparison between algorithms, we have set in each run k to the number of groups of the expected (i.e., non avoided) clustering. To tune w (the weight of the soft constraints) we have used a crossvalidation strategy, which involved testing the possible values in dataset (i) and taking the one with best results ($w = 0.0025$), using that value in the other dataset. Also, the convergence condition is tested comparing the centroids of the present iteration with those of the previous one. The process is stopped as well if a certain number of iterations is exceeded without convergence.

Table 1. Results for the avoiding bias experiment with the defined datasets for batch k-means, the new constrained k-means working with soft constraints (SCKM) and the CIB based method

Dataset (i)	Avoiding Topic (k=4)			Avoiding University (k=5)		
	MI(Topic)	Mi(Univ.)	P(Univ.)	MI(Univ.)	MI(Topic)	P(Topic)
CIB	0.0067	0.0189	0.2917	0.0085	0.2342	0.4735
Batch k-means	0.5177	0.2111	0.4395	0.3217	0.5164	0.6730
SCKM (w=0.0025)	0.0039	0.2947	0.5061	0.0031	0.4686	0.6431

Dataset (ii)	Avoiding Topic (k=2)			Avoiding Region (k=2)		
	MI(Topic)	MI(Region)	P(Region)	MI(Region)	MI(Topic)	P(Topic)
CIB	0.0015	0.0107	0.5516	0.0001	0.8548	0.9781
Batch k-means	0.0073	0.0814	0.8253	0.0965	0.0081	0.9838
SCKM (w=0.0025)	0.0003	0.1408	0.8253	0.0004	0.0054	0.9838

In Table 1 we show the results achieved by CIB, our algorithm and a batch k-means in this experiment. As in the last two algorithms the outcome of the clustering process is very dependant on the initial seeds the results shown are the average of 10 random seed initialisations. In each of these initialisations we have as well randomised the order in which the documents were inspected.

As a previous note we should stress how the MI values of the runs of the batch k-means in the datasets show unequivocally the tendency of that algorithm to one of the possible clusterings of the data, showing a real-world example where having a way to avoid that bias could come in handy.

With the trained w our algorithm performed really well, achieving the two aims of the Avoiding Bias task. Firstly, we have been able to avoid the known organisation of the data, which is visible in the considerable decrease of the values of MI for the known clustering of our algorithm and batch k-means. Secondly, the outcome of our clustering algorithm resembles more the not known organisation of the data than the known one, which can be confirmed comparing the MI for the known and unknown clustering. Moreover, in all cases the quality of the clustering (the purity for the not known clusterisation) is still high.

Comparing with the results of Gondek and Hoffman (CIB), our algorithm achieves in almost all cases noticeable increases in the similarity to the unknown clustering than their approach, with also more quality. The only exception happens in dataset (ii) when trying to avoid the "Region" criterion. This can attributed to the special nature of this dataset, which is extremely unbalanced. Nevertheless, we must stress that even in this extreme case the algorithm is able to fulfil the two aims previously pointed out.

Experiment 2: Incremental Behaviour. We have used dataset (i) to compare the behaviour of the soft and absolute negative constraints as their number is increased. Now we are not trying to avoid any clustering, but to achieve the maximum similarity (measured with RI) with the ground truth (the University criterion). The constraints were defined over nine tenths of the documents, taking randomly pairs of documents belonging to different clusters. We used this crossvalidation strategy, similar to the one used in [7], to see the direct influence of the constraints on the whole collection and the indirect influence over the non constrained documents. The results showed that, although with few constraints (< 2000) the behaviour of absolute and soft constraints is similar, improving slighty the results of batch k-means, increasing the number of absolute constraints entails a decrease of the effectiveness, well below batch k-means, a situation which does not arise with the soft constraints, which experiment a linear improvement with the number of constraints. So it has been demonstrated that in this kind of problems the soft constraints outperform the absolute constraints, which are not adequate when working with more than a few constraints.

4 Related Work

The way in which the soft constraints are introduced in our algorithm is similar to the one presented in [4] by Yang and Callan. However, they use the constraints in an algorithm specially tailored for the task of near duplicate detection. Also the algorithm only used the Must, Cannot and "Family" (similar to May) rules and they are only bidirectional. Another key difference is that their algorithm does not take advantage of the information from the previous iteration.

Also in the field of IR Ji et al. presented in [5] a semi-supervised clustering method based on spectral clustering that is very effective, but only allows the inclusion of background knowledge through soft pairwise relations of membership to the same cluster. The method is quite time consuming, as it implies the calculus of the eigenvectors of the document matrix. In [8] Klein et al. present a constrained hierarchical clustering including Must and Cannot Links. The algorithm has the problem of the computational cost of the hierarchical methods but it outperforms the Wagstaff et al. method in terms of effectiveness. However, they only evaluated it in synthetic and very small non-textual collections.

Several papers were presented recently in DM forums; one of them was the mentioned seminal paper in finding alternative clustering presented by Gondek and Hofmann [9]. They introduce an approach that uses the Conditional Information Bottleneck theory using a dual objective function searching for both alternative and good clustering. One problem of this technique is that it requires a joint distribution information for each variable and that is not always available. In [10] Bae and Bailey presented a constrained clustering method, enabling the Cannot-Link rules, based on a average-link algorithm. Although it outperformed CIB, the algorithm complexity makes it inefficient for large collections.

Some papers approach the inclusion of the constraints through the learning of distance functions [16], such as Davidson and Qi [11], which uses Must-Link and Cannot-Link knowledge but implies the use of Singular Value Decomposition (SVD), or Cui et al. [12], an approach to produce multiple orthogonal clustering views using Principal Component Analysis (PCA).

5 Conclusions

In this paper we have presented a general algorithm for constrained clustering extending the well-known constrained k-means [7] with soft-constraints. With this inclusion we still have a clustering algorithm with performance and able to work with large text collections. The new soft-constraints allow tackling the task of avoiding bias and outperform the CIB-based method [9], specially designed for that task. Our algorithm also presents a good behaviour when the number of constraints is reduced, sharing this property with other algorithms more expensive computationally like the CCL [8], and it does not degrade the effectiveness when increasing the amount of constraints but the opposite.

Acknowledgements: This work was co-funded by FEDER, SEUI and Xunta de Galicia under projects TIN2008-06566-C04-04 and 07SIN005206PR and FPU grant AP2007-02476.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31**(3) (1999) 264–323
2. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34**(1) (2002) 1–47
3. Basu, S., Davidson, I., Wagstaff, K.: Constrained Clustering: Advances in Algorithms, Theory, and Applications. Chapman & Hall/CRC (2008)
4. Yang, H., Callan, J.: Near-duplicate detection by instance-level constrained clustering. In: Proc. of SIGIR 2006 pp. 421–428
5. Ji, X., Xu, W.: Document clustering with prior knowledge. In: Proc. of SIGIR 2006 pp. 405–412
6. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: Proc. of ICML 2000 pp. 1103–1110
7. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proc. of ICML 2001 pp. 577–584
8. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proc. of ICML 2002 pp. 307–314
9. Gondek, D., Hofmann, T.: Non-redundant data clustering. In: Proc. of ICDM 2004 pp. 75–82
10. Bae, E., Bailey, J.: COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In: Proc. of ICDM 2006 pp. 53–62
11. Davidson, I., Qi, Z.: Finding alternative clustering using constraints. In: Proc. of ICDM 2008 pp. 773–778
12. Cui, Y., Fern, X.Z., Dy, J.G.: Non-redundant multi-view clustering via orthogonalization. In: Proc. of ICDM 2007 pp. 133–142
13. McQueen, J.: Some methods for classification and analysis of multivariate observations. Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability **1** (1967) 281–297
14. Pantel, P., Lin, D.: Document clustering with committees. In: Proc. of SIGIR 2002 pp. 199–206
15. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval Cambridge University Press (2008)
16. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems 15, MIT Press (2003) 505–512