Semi-supervised Clustering with Example Clusters

Celine Vens[1], Bart Verstrynge[1] and Hendrik Blockeel[1,2]
[1]*Department of Computer Science, KU Leuven, Leuven, Belgium*
[2]*Leiden Institute of Advanced Computer Science, Universiteit Leiden, Leiden, The Netherlands*

Keywords: Clustering, Semi-supervised Clustering, Constraint-based Clustering, Metric Learning.

Abstract: We consider the following problem: Given a set of data and one or more examples of clusters, find a clustering of the whole data set that is consistent with the given clusters. This is essentially a semi-supervised clustering problem, but different from those that have been studied until now. We argue that it occurs frequently in practice, but despite this, none of the existing methods can handle it well. We present a new method that specifically targets this type of problem. We show that the method works better than standard methods and identify opportunities for further improvement.

1 INTRODUCTION

We consider the task of (non-hierarchical) clustering: given a dataset D, partition D into clusters such that instances within the same cluster tend to be similar, and instances in different clusters dissimilar.

This task is usually considered unsupervised. Recently, however, some research has focused on semi-supervised clustering. Here, some information is given about which elements belong to the same cluster, usually in the form of pairwise constraints: two particular instances should be in the same cluster (must-link constraint), or should not be (cannot-link constraint). Such background information helps the system find a clustering that meets the user's criteria.

There are multiple reasons why a user might want to provide partial supervision. They all boil down to the fact that clustering is essentially underconstrained: there may be many "good" clusterings in a data set. For instance, hierarchical clustering methods yield clusterings at many different levels of granularity, and it is up to the user to select the desired level. Also, in high-dimensional spaces, a different cluster structure may occur in different subspaces, and the clustering system cannot know which subspace is the most relevant one for the user (Agrawal et al., 2005).

In this paper, we introduce a new type of semi-supervised clustering. Supervision here consists of providing one or more example clusters. This type of supervision is often quite natural. Consider entity resolution in a database of authors: the task is to cluster occurrences of author names on papers such that occurrences are in the same cluster if they refer to the same actual person.[1] If one person indicates all the papers she authored, that set of papers is an example cluster. Knowing one, or a few, such clusters may help the system determine what kinds of clusters are good, so it can better cluster the other instances.

Example clusters can be translated to pairwise constraints, but that induces many of those, distributed unevenly over the instance space. Most existing systems expect the pairwise constraints to be distributed more evenly, and have been evaluated under this condition. It is therefore not obvious that they will work well in the new setting.

This paper is a first study of this new type of semi-supervised clustering task. We first briefly survey the work on semi-supervised clustering (Section 2). We next discuss the new setting, relate it to existing settings, and argue that none of the existing methods are very suitable for this task (Section 3). We propose a novel method that focuses specifically on this task (Section 4), and experimentally evaluate it in Section 5. We conclude in Section 6.

2 SEMI-SUPERVISED CLUSTERING

Most research on semi-supervised clustering has fo-

[1]This task is not trivial because different persons may have the same name, and the same person may be referred to in different ways, e.g., "John Smith", "J.L. Smith".

Vens C., Verstrynge B. and Blockeel H..
Semi-supervised Clustering with Example Clusters.
DOI: 10.5220/0004547300450051
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (KDIR-2013), pages 45-51
ISBN: 978-989-8565-75-4

cused on providing **pairwise constraints** to the clustering algorithm. Wagstaff et al. (2000, 2001) define must-link and cannot-link constraints, respectively, for specifying that two instances should, or should not, be in the same cluster.

One way of dealing with these pairwise constraints is adapting existing clustering algorithms to take them into account. Wagstaff et al. (2001) adapt K-Means to this effect, treating must-link and cannot-link as hard constraints.

Alternatively, one can use a standard algorithm, but adapt the distance metric. Xing et al. (2002) propose to learn a Mahalanobis matrix M (Mahalanobis, 1936), which defines a corresponding Mahalanobis distance

$$d_M(x,y) = \sqrt{(x-y)^T M(x-y)} \quad (1)$$

They find the M that minimizes the sum of squared distances between instances that must link, under the constraint that $d_M(x,y) \geq 1$ for all x and y that cannot link. M can be restricted to be a diagonal matrix, or can be full. The idea of learning such a distance function is generally referred to as metric-based or similarity-adapting methods (Grira et al., 2004).

Combining algorithm and similarity adaptation, Bilenko et al. (2004) introduced the MPCK-Means algorithm. A first difference with Wagstaff et al. is that constraints are now handled in a soft-constrained manner by defining costs for unsatisfied constraints. Furthermore, the k means are initialised using a seeding procedure proposed by Basu et al. (2002). For the metric-based part of MPCK-Means, a separate Mahalanobis metric can be learned for each tentative cluster in every iteration of the algorithm, allowing clusters of different shapes in the final partition.

Alternatively to pairwise constraints, Bar Hillel et al. (2005) use **chunklets**, groups of instances that are known to belong to the same cluster. Their Relevant Component Analysis (RCA) algorithm takes chunklets as input and learns a Mahalanobis matrix. This approach is shown to work better than Xing et al.'s in high dimensional data. A downside is that only must-link information is taken into account. There is no information about which instances cannot link: different chunklets may belong to the same cluster, or they may not. RCA minimizes the same function as Xing et al.'s method, but under different constraints (Bar-Hillel et al., 2005).

Yeung and Chang (2006) have extended RCA to include cannot-link information. They treat each pairwise constraint as a chunklet, and compute a separate matrix for the must-link constraints, A_{ML}, and for the cannot-link constraints, A_{CL}. The data are then transformed by $A_{CL}^{1/2} \cdot A_{ML}^{-1/2}$. This "pushes apart" cannot-link instances in the same way that must-link instances are drawn together.

3 CLUSTERS AS EXAMPLES

3.1 Task Definition

We define the task of semi-supervised clustering with example clusters as follows (where $\mathcal{P}(\cdots)$ denotes the power set):

Given: An instance space X, a set of instances $D \subseteq X$, a set of disjoint example clusters $E \subseteq \mathcal{P}(D)$, and a quality measure $Q : \mathcal{P}(D) \times \mathcal{P}(D) \to \mathbb{R}$.

Find: A partition $\mathbf{C} = \{C_1, C_2, ..., C_k\}$ over D that maximizes $Q(\mathbf{C}, E)$.

Q typically measures to what extent $\mathbf{C}$ is consistent with E (ideally, $E \subseteq \mathbf{C}$), but may also take general clustering quality into account. Note that the number of clusters to be found, or the distance metric to be used, are not part of the input. Also, the requirement that $E \subseteq \mathbf{C}$ is not strict; this allows for noise in the data.

The task just defined has many applications. We mentioned, earlier on, entity resolution. A similar task is face recognition, in a context where all (or most) occurrences of the face of a few persons have been labeled. This application is typically high-dimensional. Clustering in high-dimensional spaces is difficult because multiple natural clusterings may occur in different subspaces (Agrawal et al., 2005). For instance, one might want to cluster faces according to identity, poses, emotions shown, etc. An example cluster can help the system determine the most relevant subspace.

3.2 Translation to Pairwise Constraints

Example clusters can easily be translated into pairwise constraints. Let $ML(x,y)$ denote a must-link constraint between x and y, and $CL(x,y)$ a cannot-link constraint. Providing an example cluster C corresponds to stating $ML(x,y)$ for all $x,y \in C$ and $CL(x,y)$ for all $x \in C, y \notin C$. If C has n elements (call them x_1, ..., x_n), and the complete dataset has N elements (x_1 to x_N), this generates $n(n-1)/2$ must-link constraints and $n(N-n)$ cannot-link constraints[2]. Clearly, this set of constraints can be large ($O(nN)$), potentially

[2]By applying two inference rules: $\forall x,y,z : ML(x,y) \wedge ML(y,z) \Rightarrow ML(x,z)$ and $\forall x,y,z : CL(x,y) \wedge ML(y,z) \Rightarrow CL(x,z)$, it suffices to list a minimal set of $n-1$ must-link constraints and $N-n$ cannot-link constraints. However, the existing metric learning methods that use pairwise constraints do not automatically apply these rules.

making existing methods slow, and the constraints are distributed unevenly, making these methods uninformed about large parts of the instance space.

Cluster examples can also be seen as "complete" chunklets. RCA could therefore be applied without any translation, but then the information about the maximality of the chunklets is lost; RCA allows separate chunklets to end up in the same cluster, which is not wanted when chunklets are known to be complete. Yeung and Chang's extension allows for negative information, which solves this problem; but this negative information is again expressed by means of pairwise constraints. Thus, although the chunklet based methods provide a concise representation for the must-link constraints, they do not provide one for cannot-link constraints, so they, too, suffer from the problem of generating many constraints.

4 CLUSTERING USING EXAMPLES: CLUE

We now present CLUE (Clustering Using Example clusters). Given some example clusters, CLUE tries to find a good overall clustering consistent with them. The proposed solution does not require the number of clusters as input.

Algorithm 1: The CLUE algorithm.

Input:
D: a data set
E: a set of example clusters $\{E_i\}_{i=1}^{k}$ with $E_i \subseteq D$
Output: a partition P of D
Algorithm:
1. Rescale all attributes linearly to [0,1]
2. Learn a Mahalanobis distance d_M that is maximally consistent with the constraints
3. Construct a dendrogram Δ by applying a bottom-up hierarchical clustering procedure with d_M
4. Find the range of partitions in Δ for which the examples clusters are reconstructed optimally
5. Within that range, find the best partition P

The high-level algorithm is shown as Algorithm 1. We next explain all steps in detail.

Step 1: Rescaling. Attributes are rescaled to $[0,1]$ to avoid the effects of incomparable ranges.

Step 2: Metric Learning. This step computes a distance metric over the instance space that best corresponds to the example clusters given. Our approach is based on Yeung and Chang (2006). We compute two matrices, A_{ML} and A_{CL}, as follows:

$$A_{ML} = \frac{1}{N_a} \sum_{E_i \in E} \sum_{x \in E_i} (x - \bar{E}_i)(x - \bar{E}_i)^T \qquad (2)$$

$$A_{CL} = \frac{1}{N_b} \sum_{E_i \in E} \sum_{x \notin E_i} (x - \bar{E}_i)(x - \bar{E}_i)^T \qquad (3)$$

with $N_a = \sum_{E_i \in E} |E_i|$ and $N_b = |D| \cdot |E| - \sum_{E_i \in E} |E_i|$.

Thus, while Yeung and Chang use as chunklets the pairwise constraints, we use as "positive" chunklets the example clusters, and as negative chunklets, pairs $(x, \bar{E}_i)$ with $x \notin E_i$ and $\bar{E}_i$ the mean of the cluster. This makes the computation of A_{ML} $O(n)$ (with n the number of instances in all example clusters together), and that of A_{CL} $O(kN)$, with N the total number of instances and k the number of example clusters. $A_{ML}^{-1/2}$ transforms the space so that examples in a cluster are drawn closer to its center (in other words, intra-cluster variance is reduced) and $A_{CL}^{1/2}$ transforms it such that examples outside a cluster are pushed farther from its center (inter-cluster variance is increased). The Mahalanobis matrix corresponding to this transformation is $M = A_{ML}^{-1/2} \cdot A_{CL} \cdot A_{ML}^{-1/2}$.

Step 3: Hierarchical Clustering. A standard bottom-up hierarchical clustering method is used, using d_M as distance metric, and using single or complete linkage. This gives a dendrogram that represents N partitional clusterings, from N singletons at the bottom to a single cluster at the top.

Step 4: Filtering the Partitions. After the bottom-up clustering procedure, we investigate all resulting partitions P_i and select those where the example clusters have been reconstructed optimally. For this, we propose the *COnstraint based Rand Index* (CORI). It is based on the Rand index measure (Rand, 1971), which is used to compare a predicted clustering to a target clustering.

Let S_{ML} and S_{CL} respectively denote the set of all must-link and cannot-link constraints induced by the example clusters. The CORI for a clustering $C = \{C_1, ..., C_k\}$ is defined as follows:

$$\text{CORI}(C) = \left(\frac{|S_{ML}^{correct}|}{|S_{ML}|} + \frac{|S_{CL}^{correct}|}{|S_{CL}|} \right) / 2 \qquad (4)$$

$$S_{ML}^{correct} = \{ML(x,y) \in S_{ML} \mid \exists i : x \in C_i \wedge y \in C_i\}$$

$$S_{CL}^{correct} = \{CL(x,y) \in S_{CL} \mid \exists i, j \neq i : x \in C_i \wedge y \in C_j\}$$

The CORI equals 0.5 at the start and the end of the agglomerative clustering procedure. Initially, all cannot-link constraints and none of the must-link constraints are fulfilled, as the instances each belong to a singleton cluster. As the clustering process advances,

the cannot-link component in the CORI definition decreases, and the must-link component increases. Finally, all must-link constraints and none of the cannot-link constraints are fulfilled when only a single cluster remains. Note that the must-link and cannot-link components of the CORI are weighted independently, in contrast to the Rand index. This is because usually $|S_{CL}| \gg |S_{ML}|$.

The result of this step is a range of clusterings with maximal CORI.

Step 5: Final Partition. In the previous step, clusterings were evaluated based on the reconstruction of the example clusters, resulting in a range of optimal clusterings. We now select from that range the clustering that yields the best overall cluster quality. Category utility (Fisher, 1987) is an evaluation metric that judges cluster quality in terms of intra- and inter-cluster dissimilarity. Witten et al. (2011) provide a definition for numeric data, by assuming normally distributed data:

$$\mathrm{CU}(C) = \frac{1}{k} \sum_{C_l \in C} Pr(C_l) \frac{1}{2\sqrt{\pi}} \sum_{i=1}^{d} \left(\frac{1}{\sigma_{il}} - \frac{1}{\sigma_i} \right), \quad (5)$$

where k is the number of clusters, d is the number of attributes, σ_i denotes the standard deviation of attribute i, and σ_{il} the standard deviation of attribute i for instances in cluster l.

Category utility gives an equal importance to all attributes. However, as the clusters were constructed using supervised information, we want to return the best clustering according to the learned distance measure. By defining a *weighted* variant of the category utility, we can interpret the dissimilarities using the Mahalanobis matrix M:

$$\mathrm{WCU}(C) = \frac{1}{k} \sum_{C_l \in C} Pr(C_l) \frac{1}{2\sqrt{\pi}} \sum_{i=1}^{d} M_i^{1/2} S_{(l)} \quad (6)$$

$$S_{(l)} = \left[\left(\frac{1}{\sigma_{1l}} - \frac{1}{\sigma_1} \right) \cdots \left(\frac{1}{\sigma_{dl}} - \frac{1}{\sigma_d} \right) \right]^T$$

where $M_i^{1/2}$ denotes the i-th row of $M^{1/2}$.

5 EVALUATION

In this section, we empirically evaluate our approach towards learning from example clusters on synthetic and real world datasets. We compare our method - both with a single (SL) and complete linkage (CL) for the agglomerative clustering step - with K-MEANS and MPCK-MEANS (Bilenko et al., 2004) (see Section 2). Since MPCK-MEANS can learn either one global or multiple local distance metrics, we tested both cases. K-MEANS and MPCK-MEANS are provided with the exact number of clusters as input.

5.1 Datasets

5.1.1 Synthetic Data

A synthetic dataset was created with 200 instances and 6 numeric dimensions with different domain sizes, see Figure 1. Three dimensions were randomly generated, one dimension contains five bar-shaped clusters, and two dimensions together form 16 circle-shaped clusters. These are the two target clusterings that we try to discover using an example cluster.

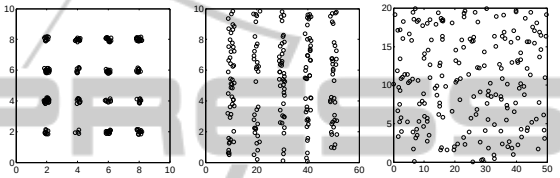

Figure 1: The 6 dimensions of the synthetic dataset.

5.1.2 Real-world Data

We used three UCI datasets (Frank and Asuncion, 2010). **CMU Face Images** contains 640 pictures of 20 different persons[3], each shown with 4 poses, 4 emotions, and with or without sunglasses. This is a nice example of a dataset that can naturally be clustered in several ways. We used the "identity" and "pose" as target clusterings. Principal Component Analysis was applied to the original data to represent the images as linear combinations of eigenfaces (Turk and Pentland, 1991). Only the first 100 eigenfaces were kept; this allowed us to represent the data in a more compact way, while presevering 97% of the original variance in the data. **Libras Movement** contains 15 classes of 24 instances each. Each class references to a hand movement type in Brazilian signal language. **Seeds** contains measurements of seven geometrical properties of kernels belonging to three different varieties of wheat. It has 210 instances.

5.2 Evaluation Measures

In our evaluation, we wish to compare the returned clustering to a target clustering. We now discuss the measures we will use for this. In the following, $C_p = \{C_{p1}, C_{p2}, ..., C_{pk}\}$ denotes the predicted clustering, and $C_t = \{C_{t1}, C_{t2}, ..., C_{tl}\}$ the target clustering. The function $p : X \to \{1, ..., k\}$ returns the predicted

[3]Due to a corrupted image file, the identity "karyadi" was left out, resulting in 19 identities.

cluster for an instance, and similarly, $t : X \to \{1,...,l\}$ returns the target cluster. The notations $p(x) = i$ and $t(x) = j$ are abbreviated as $p_i(x)$ and $t_j(x)$.

Rand Index. The Rand Index (RI) is a popular measure for comparing clusterings. It expresses the proportion of instance pairs for which both clusterings agree on whether they are in the same cluster or not. When there are many clusters, RI can be dominated by instances correctly predicted not to be in the same cluster. For instance, if $C_p = \{\{a,b\},\{c,d\},\{e,f\},\{g,h\},\{i,j\}\}$ and $C_t = \{\{j,a\}, \{b,c\},\{d,e\},\{f,g\},\{h,i\}\}$, we obtain $RI = \frac{0+35}{45} = \frac{7}{9}$, a high score for a very bad predicted clustering.

Normalized Mutual Information. Normalized Normalized mutual information (NMI) (Manning et al., 2008) measures the amount of information that is shared by two clusterings, and penalizes large clusterings. It is defined as follows:

$$NMI(C_p;C_t) = \frac{MI(C_p;C_t)}{(H(C_p)+H(C_t))/2} \qquad (7)$$

$$MI(C_p;C_t) = \sum_{i=1}^{k}\sum_{j=1}^{l} Pr(p_i(x),t_j(x)) \cdot \log\left(\frac{Pr(p_i(x),t_j(x))}{Pr(p_i(x)) \cdot Pr(t_j(x))}\right)$$

$$H(C_p) = -\sum_{i=1}^{k} Pr(p_i(x)) \cdot \log\left(Pr(p_i(x))\right)$$

However, this measure can give unexpected results if the individual cluster cardinalities differ substantially. For instance, consider $C_t = \{\{a\},\{b\},\{c\},\{d,e,f\}\}$ and $C_p = \{\{a,b,c\},\{d,e,f\}\}$. If we add more and more instances to the last cluster in both C_t and C_p, then $H(C_p)$ and $H(C_t)$ will get closer to zero, making C_p a better clustering, although it still only correctly finds one of the four clusters.

Complemented Entropy. To deal with the above shortcomings, we propose a new clustering evaluation measure, called complemented entropy (CE). It scores the entropy of the target labels in the predicted clusters (H_t), as well as the entropy of the predicted labels in the target clusters (H_p). These entropies are in a sense complementary: predicting too few clusters will increase H_t, while predicting too many clusters will increase H_p. A formal definition is given below:

$$H_t = -\sum_{i=1}^{k}\sum_{j=1}^{l} Pr(t_j(x) \mid p_i(x)) \cdot \log\left(Pr(t_j(x) \mid p_i(x))\right)$$

$$H_p = -\sum_{j=1}^{l}\sum_{i=1}^{k} Pr(p_i(x) \mid t_j(x)) \cdot \log\left(Pr(p_i(x) \mid t_j(x))\right)$$

$$CE = 1 - \left(\frac{H_t}{maxH_t} + \frac{H_p}{maxH_p}\right)/2 \qquad (8)$$

In this definition, $maxH_t$ denotes the maximally possible value for H_t and is reached when all predicted clusters contain an equal number of target labels, and analogously for $maxH_p$. Consider again the previous example. If we increase the number of instances in the last clusters, leading to $C'_t = \{\{a\},\{b\},\{c\},\{d,e,f,g,h,i,j\}\}$ and $C'_p = \{\{a,b,c\},\{d,e,f,g,h,i,j\}\}$, then the CE score remains unchanged (while NMI went up). The CE can only become one when the predicted clustering perfectly matches the target clustering.

NMI vs. CE. NMI and CE evaluate different aspects of the predicted clustering. Roughly, NMI gives more weight to clusters with high cardinality, while CE treats all clusters equally. Figure 2 illustrates the difference for two clusterings over the same dataset.

(a) NMI=0.89, CE=0.84 (b) NMI=0.66, CE=0.91

Figure 2: NMI prefers the clustering where more instance pairs are labeled consistently (left), CE prefers the one where more target clusters are reconstructed (right).

5.3 Experimental Results

The experiments are set up as follows. In each run, we use one target cluster as an example cluster, run the clustering method, and compute the RI, NMI and CE for the part of the clustering that excludes the example cluster. For each dataset, we repeat this procedure for each target cluster and report the average and standard deviation of the results. For MPCK-MEANS, the example cluster is translated into a set of must-link and cannot-link constraints. K-means consistently ignores the constraints.

Table 1 presents the results for NMI, CE, and RI. We observe that 18 out of 22 highlighted results are in CLUE rows (9 for CLUE-CL, 9 for CLUE-SL); MPCK-MEANS(global) scores 4, the others 0.

Surprisingly, MPCK-MEANS scores worse than (unsupervised) K-MEANS in about half of the cases, which suggests that using non-evenly spread constraints may actually hurt its performance. CLUE has a similar issue on one dataset (Seeds).

We also observed (not shown here) that CLUE returns too many clusters. (The other systems use the number of clusters as an input, so they cannot go wrong there.) This turns out to be a result of overfitting: the learned Mahalanobis distance compresses

Table 1: Results.

	Synthetic (Bars)			Synthetic (Circles)		
Method	NMI	CE	RI	NMI	CE	RI
K-MEANS	0.039 (0.005)	0.051 (0.005)	0.651 (0.003)	0.501 (0.006)	0.529 (0.008)	0.896 (0.001)
MPCK-MEANS_glob	0.757 (0.247)	0.757 (0.246)	0.878 (0.125)	0.281 (0.033)	0.336 (0.031)	0.866 (0.005)
MPCK-MEANS_loc	0.391 (0.206)	0.513 (0.121)	0.670 (0.206)	0.252 (0.050)	0.419 (0.086)	0.806 (0.129)
CLUE, SL	**1.000** (0.000)	**1.000** (0.000)	**1.000** (0.000)	0.883 (0.115)	0.932 (0.079)	0.937 (0.061)
CLUE, CL	**1.000** (0.000)	**1.000** (0.000)	**1.000** (0.000)	**0.952** (0.084)	**0.965** (0.062)	**0.984** (0.022)
	Faces (Identity)			Faces (Pose)		
Method	NMI	CE	RI	NMI	CE	RI
K-MEANS	0.717 (0.013)	0.768 (0.008)	0.930 (0.004)	0.039 (0.012)	0.133 (0.016)	0.561 (0.007)
MPCK-MEANS_glob	**0.797** (0.024)	0.829 (0.020)	**0.944** (0.007)	0.035 (0.027)	0.074 (0.042)	0.555 (0.012)
MPCK-MEANS_loc	0.673 (0.014)	0.711 (0.013)	0.924 (0.003)	0.026 (0.012)	0.054 (0.024)	0.554 (0.004)
CLUE, SL	0.725 (0.199)	**0.892** (0.046)	0.845 (0.208)	**0.357** (0.021)	**0.637** (0.019)	**0.681** (0.012)
CLUE, CL	0.706 (0.126)	0.780 (0.086)	0.911 (0.097)	**0.357** (0.013)	0.619 (0.011)	0.675 (0.003)
	Libras			Seeds		
Method	NMI	CE	RI	NMI	CE	RI
K-MEANS	0.559 (0.011)	0.615 (0.008)	0.899 (0.003)	0.641 (0.143)	0.704 (0.032)	0.853 (0.063)
MPCK-MEANS_glob	0.535 (0.021)	0.558 (0.026)	0.888 (0.004)	**0.750** (0.218)	0.760 (0.209)	**0.899** (0.089)
MPCK-MEANS_loc	0.433 (0.015)	0.534 (0.017)	0.857 (0.004)	0.748 (0.218)	0.764 (0.205)	0.891 (0.096)
CLUE, SL	0.641 (0.006)	**0.746** (0.017)	0.931 (0.005)	0.380 (0.254)	**0.768** (0.081)	0.659 (0.156)
CLUE, CL	**0.645** (0.009)	0.744 (0.008)	**0.933** (0.002)	0.453 (0.237)	0.631 (0.177)	0.679 (0.134)

the example cluster very well, but the other clusters much less. As a result, the example cluster is reconstructed well before other clusters are. This overfitting seems inherent to the rescaling approach that also earlier methods use, and may explain why the existing methods can perform worse than unsupervised clustering. We are still investigating this issue.

6 CONCLUSIONS

We introduced a novel type of supervision for semi-supervised clustering. The supervision consists of one or more complete example clusters. Whereas existing semi-supervised clustering methods assume limited knowledge over the complete instance space, this setting assumes complete knowledge over a limited part of the instance space.

We have proposed a novel method designed specifically for this task. It learns a Mahalanobis distance that is maximally consistent with the given example clusters. Then it performs agglomerative clustering using this distance. Finally, it returns the partition for which the example clusters are reconstructed optimally. Evaluating this method on six clustering tasks, we have found that the novel method performs better than existing methods in this setting. The evaluation also points to a problem of "overfitting the example cluster" which is as yet unresolved.

ACKNOWLEDGEMENTS

Celine Vens is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO-Vlaanderen). Work supported by the Research Foundation - Flanders (G.0682.11) and the KU Leuven Research Fund (GOA 13/010).

REFERENCES

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (2005). Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33.

Bar-Hillel, A., Hertz, T., Shental, N., and Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965.

Basu, S., Banerjee, A., and Mooney, R. (2002). Semi-supervised clustering by seeding. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002*.

Bilenko, M., Basu, S., and Mooney, R. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pages 81–88.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

Grira, N., Crucianu, M., and Boujemaa, N. (2004). Unsupervised and Semi-supervised Clustering: a Brief Survey. *A Review of Machine Learning Techniques for*

Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6).

Mahalanobis, P. C. (1936). On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, pages 49–55.

Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, (336):846–850.

Turk, M. A. and Pentland, A. P. (1991). Face recognition using eigenfaces. *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 591(1):586–591.

Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110.

Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). Constrained K-means clustering with background knowledge. In *ICML*, pages 577–584. Morgan Kaufmann.

Witten, I., Frank, E., and Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Xing, E., Ng, A., Jordan, M., and Russell, S. (2002). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press.

Yeung, D. and Chang, H. (2006). Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. *Pattern Recognition*, 39(5):1007 – 1010.