An evaluation of filter and wrapper methods for feature selection in categorical clustering

Luis Talavera

Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona 1-3 08034 Barcelona, Spain
talavera@lsi.upc.edu

Abstract. Feature selection for clustering is a problem rarely addressed in the literature. Although recently there has been some work on the area, there is a lack of extensive empirical evaluation to assess the potential of each method. In this paper, we propose a new implementation of a wrapper and adapt an existing filter method to perform experiments over several data sets and compare both approaches. Results confirm the utility of feature selection for clustering and the theoretical superiority of wrapper methods. However, it raises some problems that arise from using greedy search procedures and also suggest evidence that filters are a reasonably alternative with limited computational cost.

1 Introduction

It is widely recognized that a large number of features can adversely affect the performance of inductive learning algorithms, and clustering is not an exception. However, while there exists a large body of literature devoted to this problem for supervised learning tasks [9, 1], feature selection for clustering has been rarely addressed. The problem appears to be a difficult one given that it inherits all the uncertainties that surround this type of inductive learning. Particularly, that there is not a single performance measure widely accepted for this task and the lack of supervision available (e.g. class labels).

Although recently there has been a growing interest in feature selection for clustering, a number of questions still remain open. Wrappers for feature selection have been recently proposed with some success. However, they exhibit some limitations. The first, and probably on of the most important deficits is the lack of a more extensive empirical evaluation of the methods and, in particular, a comparison between filters and wrappers. A second shortcoming is that many of these approaches are focused on numerical clustering, and there is no theoretical or experimental evidence related to their behavior on categorical data.

In this paper we present a first attempt to fill these gaps by comparing the performance of wrapper and filter methods over several data sets. We propose a new wrapper implementation and use a filter technique based upon previous work for the experiments.

2 Feature selection for clustering

In supervised learning, feature selection is often viewed as a search problem in a space of feature subsets. To carry out this search we must specify a starting point, a strategy to traverse the space of subsets, an evaluation function and a stopping criterion. Although this formulation allows a variety of solutions to be developed, usually two families of methods are considered. On one hand, *filter* methods use an evaluation function that relies solely on properties of the data, thus is independent on any particular algorithm. On the other hand, *wrapper* methods use the inductive algorithm to estimate the value of a given subset.

Wrapper methods are widely recognized as a superior alternative in supervised learning problems, since by employing the inductive algorithm to evaluate alternatives they have into account the particular biases of the algorithm. However, even for algorithms that exhibit a moderate complexity, the number of executions that the search process requires results in a high computational cost, especially as we shift to more exhaustive search strategies.

Implementing a wrapper is a straightforward task in supervised learning, since there always some external validation measure available. Typically, one executes a classifier and obtains an estimation of the accuracy in predicting a class label that is known. Although class label prediction can be used as an external measure to assess the validity of a clustering in rediscovering a known structure, labels are not available during the learning process, so they cannot be used in a wrapper implemention for clustering.

A solution is to assume that the goal of clustering is to optimize some objective function which helps to obtain 'good' clusters and use this function to estimate the quality of different feature subsets. Despite the unavailability of class labels, this approach seems to be more reasonable that requiring clustering algorithms to maximize accuracy over a piece of information which they do not have access to. Actually, we can view the objective function as the "accuracy" of clustering algorithms. When a given algorithm is used, there is an implicit assumption that the higher (lower) the value of its objective function the better are the properties that the groups discovered exhibit.

When using an objective function in a wrapper it must applied to clusterings obtained with subsets of features of different cardinality. Since we need to compare these results, the function must be defined in a way that is not biased with respect to the number of features, that is, it should not be monotonically increasing or decreasing as a function of the dimensionality of the data. For example, as reported in [5] the scatter separability and the maximum likelihood criteria suffer this drawback.

Filter methods appear to be a, probably less optimal, but reasonable compromise for feature selection problems. But then again, for clustering tasks this turns out to be a hard problem since we need to decide what is going to be relevant to discover a structure that we do not know in advance. As before, existing supervised approaches for filtering rely mainly in properties and relationships between the data and a predefined class label.

A particularly optimal implementation of filters are methods that employ some criterion to score each feature and provide a ranking. From this ordering, several feature subsets can be chosen, either manually of setting a threshold. This special case of the filter approach, that will be refer to as *rankers*, can be extremely efficient because is a one step process without any search involved. In practice, the efficiency depends on the computational complexity of the ranking procedure.

3 EM clustering with feature selection

In this work, we adopt a commonly used and simple probabilistic framework for clustering assuming that the data comes from a multinomial mixture model with k sources corresponding to the number of clusters ([11]). This model is closely related to the naive Bayes model for classification as it relies on the assumption that all features are rendered mutually independent by the cluster variable.

We use the EM algorithm to estimate the maximum likelihood (ML) parameters and the posterior cluster probabilities for each data point. Briefly, this algorithm is an iterative procedure that alternates between two steps: the Expectation step (E) and the Maximization step (M). In the E step for every we use the current parameters to compute the partial assignment (weights) to the k clusters for each data point. In the M step, we reestimate the parameters as the ML assignment given these weights.

There are not as many clustering algorithms for categorical data as there exists for numerical data, but still there are other possible approaches, notably Cobweb [6]. However, we made the choice of EM because it produces flat clusterings as opposed to Cobweb, which builds cluster hierarchies. We think that for adequately assessing feature selection methods, the representational bias is an important factor that should be fixed, and, currently, flat clustering algorithms are more representative. Nevertheless, most categorical clustering algorithms rely on counting and computing frequencies, so that our results within the EM framework have a good chance to generalize to other algorithms.

3.1 An EM wrapper

As previously noted, the ML criterion for cluster quality has a bias of increasing as the number of features decreases, so that it cannot be used to define a wrapper. We propose a solution that assumes that the goal of feature selection is to obtain a clustering with a reduced set of features of similar or better quality as that obtained by using all the features. Intuitively, if we build a clustering with a reduced feature set, then compute the objective function adding the rest of features and find that the resulting score is as good as the one that is obtained by using all the features, this is an indicator that the non-selected features were not relevant. Therefore, the full set log-likehood can be used to guide the search of wrapper approach. Note that this method of evaluation can be potentially applied to any objective function, not only likelihood-based approaches.

An equivalent proposition has been made in the context of feature selection for unsupervised learning of conditional Gaussian networks [13].

In our probabilistic framework, we can run the EM algorithm for a given subset and estimate the model parameters, and then compute the log-likehood that these parameters yield using the full feature set. We can estimate this score in a simple manner by running an additional M step of the *EM* algorithm in which the parameters for the removed features are estimated from the weights obtained using only the selected subset. A subsequent E step would provide the full feature set likelihood estimation.

Since using exhaustive search strategies is prohibitive, wrapper methods often resort to heuristic methods and, particularly, greedy approaches. A commonly used procedure is *sequential stepwise selection* that adds or removes a single feature at each step of the search. We can start from the full set of features and use a removal operator (*backward elimination*) or start from the empty set and add one feature at a time (*forward selection*). Since repeatedly using the clustering algorithm is already a costly solution, in this paper we resort to an implementation that combines EM with forward selection because is significantly cheaper that backward elimination. We call this implementation EM-WFS (EM wrapper with forward search).

With these assumptions we have defined an starting point, a search strategy and an evaluation function, but we also need a stopping criterion. Usually we would continue the process until no improvement on the evaluation function is found. However, we have noted that, at certain points, the change of the function scores is very small. Because of that, in our implementation we stop if the relative change of the score is less than a fixed threshold.

3.2 A (dependency-based) EM ranker

One view about the relevance of features conjectures that features that are not highly correlated with other features are not likely to play an important role in the clustering process and can be deemed as irrelevant [15]. This conjecture can be explained from two points of view.

The first view argues that a general principle common to most clustering systems is to form clusters having most feature values common to their members (*cohesion*) and few values common to members of other clusters (*distinctiveness*). These properties can be expressed in the form of the conditional probabilities $P(F_i = V_{ij} \mid C_k)$ and $P(C_k \mid F_i = V_{ij})$, where F_i is a given feature, V_{ij} is some value of this feature and C_k is a cluster. By rewarding clusterings that simultaneously maximize both probabilities for given values, at the same time, clusters formed around feature correlations are favored (see [15] for examples). Therefore, features that exhibit low dependencies with other features, are not good candidates to obtain cohesive and distinct groups and, hence, irrelevant.

A second approach stems from considering the clustering problem as *mixture modeling* in which the data is assumed as being generated from a mixture of several distributions. This approach can be encoded as a Bayesian network which contains a hidden variable corresponding the clusters in the data. A commonly

used simplification assumes that all the features are conditionally independent of every other feature given the cluster variable, so that the underlying dependency model is a Naive Bayes model. The Bayesian interpretation of this approach is that the hidden variable explains or captures the dependencies of the rest of features. Thus, the resulting clusters will be most influenced by the strongest feature dependencies in the data. Hence again, features that are least correlated with other features are likely to be good candidates to eliminate.

Formulated in either way, the assumption that feature dependences are important to determine their importance for clustering tasks is independent of any labeling of the data. Therefore, it can be employed as a foundation in designing filters for feature selection for clustering. Still, this is a very general formulation that does not indicate nor how to model these dependencies neither how to employ this information in the feature selection process.

The previous assumption relating dependency and irrelevance of features provides a guide to design filter methods in feature selection for clustering, We can score each feature with a measure reflecting the degree in which this feature is dependant of other features in the data. With such a measure, we can implement a feature selection method by constructing a rank of features and selecting the best k, where k is a user given parameter.

We will assume that we can capture feature dependencies via pairwise interactions. For instance, using a mutual information measure, we can define the score of a feature F_i to be:

$$score(F_i) = \sum_{j=1, j \neq i}^{n} I(F_i; F_j) \tag{1}$$

where $I(F_i; F_j)$ stands for the usual definition of the mutual information between two variables x and y:

$$I(x; y) = \sum_{x} \sum_{y} p(x, y) log \frac{p(x, y)}{p(x)p(y)} \tag{2}$$

A simple method can be implemented by using this measure to order the features and obtain a ranking with a $O(nm^2)$ cost, where n is the number of instances and m is the number of features. We will refer to this method as EM-PWDR (EM pairwise dependency ranker).

The straightforward implementation of a ranker lefts up to the user the task of decide the number of features selected. To provide some help in this task, we added an additional step that builds a clustering with each of the feature subsets that result from the ordering (with one feature, two features, three features and so on) and then perform a single iteration to obtain the log-likelihood over the full feature set, as explained before. This figure can be used to conjecture the behavior or different subsets, although being obtained from training data, can be somewhat optimistic.

Dataset	Instances	Attributes
vote	435	16
mushroom	8124	22
LED+17	5000	24
WDBC	569	30
ionosphere	351	34
spambase	4601	57
sonar	208	60
splice	3186	60
yeast	208	79
musk	6598	166

Table 1. Characteristics of the data sets used in the experiments.

4 Empirical evaluation

In order to compare the performance of the EM-WFS and EM-PWDR methods, we performed experiments on ten data sets from the UCI Repository. The data sets and their characteristics are listed in Table 1. Data sets including numerical features were previously discretized and missings were removed by substituting those values by the mode.

As previously described, performance is estimated by computing the log-likehood of the obtained clustering over the full feature set at the end of the process. To avoid an optimistic estimation, we applied a ten-fold cross validation procedure in order to apply the feature selection procedure over a training set and compute the log-likelihood over a separate test set.The same folds were used for each of the methods.

Since the EM algorithm can be trapped in at a local maximum, both the wrapper and the ranker used at each run of EM the best of 5 runs starting with different random weight assignments. Additionally, we made the algorithm to stop when the relative difference between the likelihoods computed in two consecutive iterations did not change by 0.0001. This constrain is justified by the fact that this algorithm tends to converge asymptotically.

Figure 1 shows the log-likehood averaged over the training and testing sets when a fixed number of features is selected for each fold. A first trend that can be observed is that feature selection does not tend to decrease the quality of the clusterings with respect to the original score using all the features. Obviously, selecting the smaller subsets drops cluster quality, but the rest of combinations consistently equal or improve the full feature set results. It appears that in some data sets using the full set of features hinders the capability of the EM algorithm to converge to a good model. This result suggest that feature selection might be even more important in clustering that in supervised learning, which makes sense, since clustering algorithms must consider a large number of possible relationships between the features.

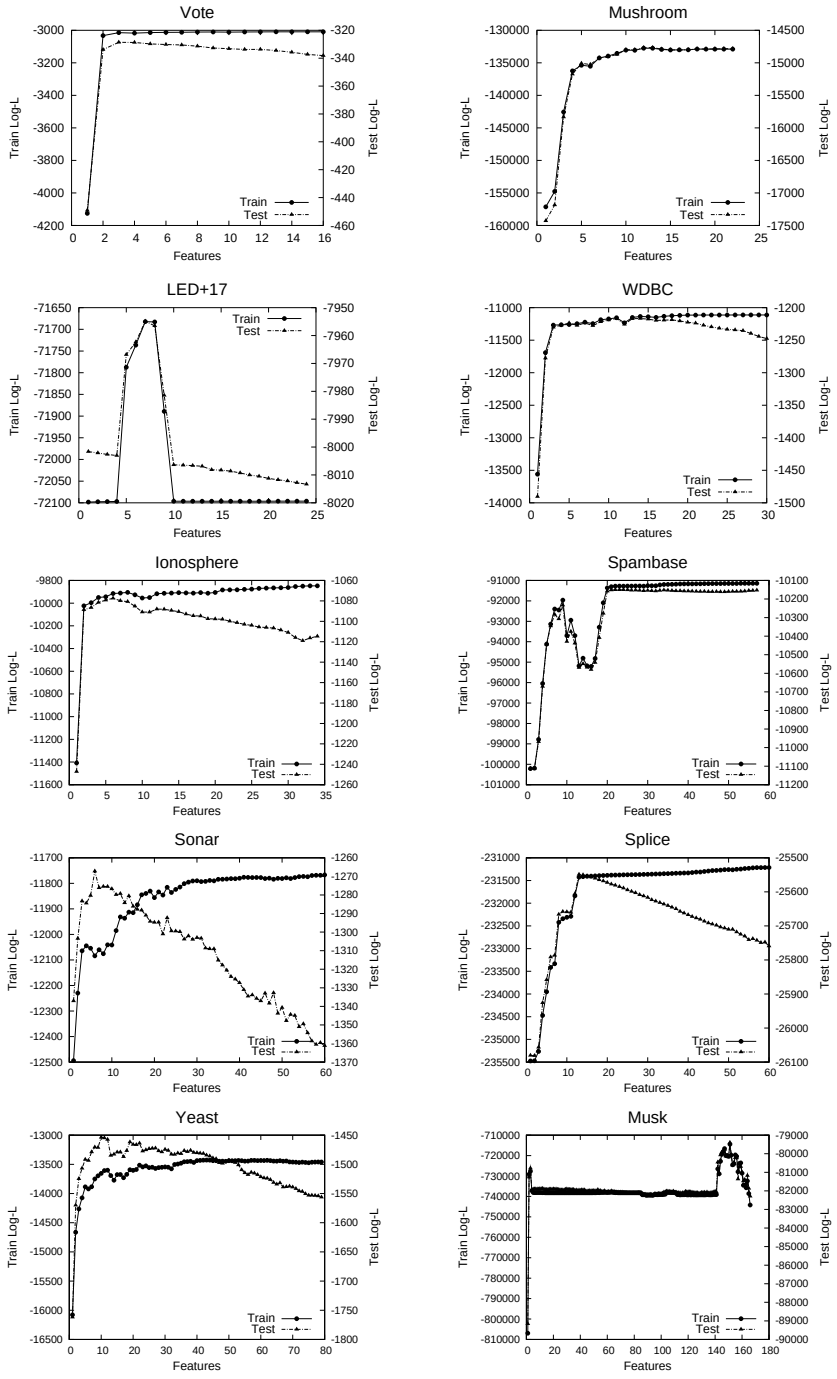

Fig. 1. Average log-likelihood over training and testing sets of EM-PWDR over different number of features.

	EM-WFS-0.001		EM-WFS-0.0001		EM-PWDR 1%		EM-PWDR-best	
Dataset	Log-L	Feat.	Log-L	Feat.	Log-L	Feat.	Log-L	Feat.
vote	-330.12	4.1	-331.40	6.1	-332.52	2	-327.77	5.3
mushroom	-15908.62	2.5	-14896.97	4.5	14907.50	8	-14750.90	15.3
LED+17	-8001.47	1	-8001.47	1	-7954.89	7	-7953.23	3
wdbc	-1215.62	4.6	-1217.71	8.4	-1218.27	7.9	1209.70	11.10
ionosphere	-1089.65	5.5	-1092.57	12.4	-1079.29	5	-1075.87	5.7
spambase	-10619.14	3.8	-10138.22	15.8	-10233.10	9	-10143.42	29.8
sonar	-1247.05	4.5	-1255.77	12.1	-1287.79	16	-1255.40	11.4
splice	-26080.95	1	-26080.95	1	-25791.47	6	-25543.38	11.1
yeast	-1437.85	9	-1437.40	14.3	-1473.25	25	-1440.11	13.9
musk	-88381.39	1	-88185.38	1.9	-79399.90	151	-77563.88	92.6

Table 2. Average test log-likelihood for different stopping criteria of EM-WFS, EM-PWDR with heuristic selection of the number of features and the best result of EM-PWDR.

A second, possibly surprising, trend that some data sets exhibit is that performance on training data is a good predictor of performance on unseen test data. Particularly on the vote, mushroom, LED+17, spambase and musk data sets the overlapping is close to perfect. And in most of the rest, even differing to some extend, training performance still can be used as a guide to select a reasonably good subset. Note that if, instead of selecting the subset with maximum training quality, we allow a deviation from the maximum, we still can obtain impressive results even with those data sets.

Table 2 shows the results for the EM-WFS method with two different stopping thresholds, namely 0.001 and 0.0001. Additionally, results for a manual selection method for EM-PWDR that chooses a number of features based on the maximum likelihood over training data is also shown. To avoid overfitting, we allow a 1% deviation from the maximum quality observed in the curve. The final column lists the best possible selection that could be made for the EM-PWDR. As expected, the wrapper performs well and somewhat better in general than the ranker. There are times where EM-PWDR could obtain a similar result but at the expense of selecting more features. However, most of the times the quality decreases by a relative factor under 1%. Moreover, in three data sets EM-WFS gets trapped in a local maximum, selecting too few features and producing unsatisfactory results.

As we could expect, wrapper methods are significantly more expensive than filter ones. In order to develop a machine independent measure of complexity, we will consider a more abstract measure than running times based upon the number of required feature comparisons. The EM algorithm exhibits a complexity $O(mnk)$ in each iteration for n instances, m features and k clusters. Therefore, we assume that a single execution of the algorithm performs $mnkI$ feature comparisons. On the other hand, the ranking method requires to compute the mutual information $(m(m-1)n)/2$ times. Note than in order to simulate a man-

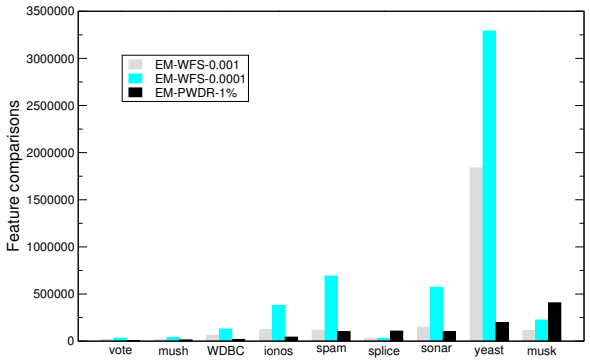

Fig. 2. Relative computational cost of the FS methods as a function of the number of feature comparisons.

ual selection of the number of features by plotting the curve of the likelihood on the full feature set additional runs of the EM algorithm over each subset is required with the added cost. Figure 2 shows the computational complexity for each method on each data set. With the exception of the cases in which the wrapper is trapped on local maxima, the computational cost is always more expensive than filter methods, especially as the number of features increases. Note that the repeated execution of the clustering algorithm is likely to be always an expensive procedure, since, unlike some lazy or semi-lazy supervised approaches (e.g. Naive Bayes) most batch clustering methods rely on some form of iterative optimization.

The most important advantage of using wrappers lies in the fact that, in some cases, they are able to achieve the same performance than filters with a more reduced subset. The most probable reason is that the dependency-based ranker is sensitive to redundant features. Even though we aim to find correlated features to be the core of the discovered clusters, there will be cases in which some features will not provide any improvement over the selected subset.

Summing up, we could say that the ranker does a reasonably good job given the limited information that uses and its significant lower complexity respect to the wrapper. The wrapper has the potential to make an accurate selection but experiments suggest evidence that it is too prone to get trapped in local maxima, a well known problem for forward search strategies. A more conservative backward search method or different search strategies, such as best first search [9], could be used to overcome this problem but at the price of increasing the already high complexity of the wrapper solution.

5 Related work

Although recently several works studying the problem of feature selection for clustering have appeared in the literature, filter based approaches are still un-

common. A notable exception is a proposal which develops an unsupervised entropy measure for ranking features [2–4]. Although several data sets are used in the evaluation, different assessment measures are employed in these works making difficult a direct comparison.

The dependency-based ranker presented in this paper has been previously used with success with hierarchical clusterings with alternative evaluation measures. In [15] the method is evaluated by comparing cluster predictions with ground truth labels, while in [14] the average predictive power over all the features (flexible prediction) is employed. A variant of the dependency assumption for continuous features has been presented in [13] for feature selection in learning conditional Gaussian networks.

An alternative to filter methods is to embed the feature selection task into the clustering process itself. The model based paradigm offers a natural way of achieving this goal by modeling feature relevance as parameters of the model. Examples of this approach are found in [10] and [17]. Results are, again, difficult to compare since the former work makes a very limited empirical evaluation using error rates and only numerical data, while the latter is focused on document clustering.

Early work in embedding feature selection into the clustering process traces back to early work by Gennari [7] that implemented a wrapper over the CLASSIT hierarchical clustering system, although at that time there was a limited availability of data for evaluation. The work is based upon selecting the features that most contribute to the clustering objective function, an idea that is also used in a filter proposed in [16] also for hierarchical clusterings.

Finally, wrapper approaches are found in [5] and [8]. The experimental evidence in these papers tend to focus on investigating the particular issues of the presented methods rather than on exploring the performance on a wide range of data sets. As it is the general case, evaluation is performed basically on numerical data.

6 Concluding remarks

In this work we have presented, to our knowledge, the first extensive empirical comparison between filter and wrapper methods of feature selection for clustering for categorical data. As it is the case with supervised learning approaches, feature selection can increase the quality of the results while reducing the complexity of the learning task.

As widely reported in the literature, wrapper methods tend to be superior to filters, and it appears that clustering is not an exception. However, the forward selection mechanism used in this work has not proved to be reliable enough, being too prone to stop in local maxima. This is an interesting result not mentioned in other papers using wrappers in feature selection for clustering. Although this could be a byproduct of our particular evaluation function, we think that the lack of references in other works to this undesirable behavior is the limited variety of data sets used.

Our results confirm previous work in that dependency based filters are a reasonably feature selection alternative. Interestingly, most often than not training quality has shown to be a good indicator of performance so that the resulting curves could be used as a guide to select the appropriate number of features. Our evaluation function appears to be intuitive and can be generalized to any objective function. However, future work could pursue a comparison with alternative approaches, such as the one presented in [5].

The computation of pairwise dependencies used in this work relies on the implicit assumption that all the features are independent given each other. This may not be the case, but supervised methods such as Naive Bayes that make the same assumption have been successfully used in a variety of learning tasks. Moreover, this is actually the same assumption that is made by the simple probabilistic model used in our implementation of the EM algorithm. It remains to be seen whether performance can be improving by using methods that do not assume that all features are independent of each other. More complex dependencies involving several features might exist but not be correctly reflected by these scores. In some cases, we could expect that by summing across all the features, some spurious dependencies might amplify the score thus producing a less accurate ranking. Future work could study more elaborated methods to score the dependence between features.

The previous issue migh be connected with the limitation exhibited by the ranker method in that it is unable to detect redundant features. The score computed cannot differentiate between required correlations that lead to good clusters and those that do not provide improvements on the light of the already selected features. The characterization of when a feature has to be considered redundant in clustering problems and the detection of this kind of features remains still an open issue.

An additional problem that could hinder the capabilities of filter methods is the existence of different good feature subsets that may lead to different, but of similar quality clusterings. In such a case, the feature ranking could be mixing features that are relevant in different contexts, thus yielding an unoptimal ordering. This assumption makes an interesting connection to a different area of research, subspace clustering [12] that could be worth to pursue.

Finally, it would be interesting to perform additional comparisons employing alternative filter approaches. Although there is almost no work on this area, the method suggested in [2] appears to be a good candidate.

References

1. A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.
2. M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering - a filter solution. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 115–122, Maebashi City, Japan, 2002. IEEE Computer Society.

3. M. Dash and H. Liu. Feature selection for clustering. In *Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference, PADKK 2000*, volume 1805 of *Lecture Notes in Computer Science*, pages 110–121, Kyoto, Japan, 2000. Springer.
4. M. Dash, H. Liu, and J. Yao. Dimensionality reduction for unsupervised data. In *Ninth IEEE International Conference on Tools with AI, ICTAI'97*, 1997.
5. J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.
6. D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.
7. J. H. Gennari. Concept formation and attention. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 724–728, Irvine,CA, 1991. Lawrence Erlbaum Associates.
8. Y. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6(6):531–556, 2002.
9. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.
10. M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.
11. M. Meila and D. Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42(1/2):9–29, 2001.
12. L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105, 2004.
13. J. M. Peña, J. A. Lozano, P. Larrañaga, and I. Inza. Dimensionality reduction in unsupervised learning of conditional gaussian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):590–603, 2001.
14. L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 389–397, Bled, Slovenia, 1999. Morgan Kaufmann.
15. L. Talavera. Dependency-based feature selection for symbolic clustering. *Intelligent Data Analysis*, 4(1), 2000.
16. L. Talavera. Feature selection and incremental learning of probabilistic conc ept hierarchies. In *Proceedings of the Seveteenth International Conference on Machine Learning*, pages 951–958, Stanford, CA, 2000. Morgan Kaufmann.
17. S. Vaithyanathan and B. Dom. Model selection in unsupervised learning with applications to do cument clustering. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 433–443, Bled, Slovenia, 1999. Morgan Kaufmann.