Feature Selection as State-Space Search: An Empirical Study in Clustering Problems

Julian R. H. Mariño and **Levi H. S. Lelis**
Departamento de Informática
Universidade Federal de Viçosa
Viçosa, Minas Gerais, Brazil

Abstract

In this paper we treat the problem of feature selection in unsupervised learning as a state-space search problem. We introduce three different heuristic functions and perform extensive experiments on datasets with tens, hundreds, and thousands of features. Namely, we test different search algorithms using the heuristic functions we introduce. Our results show that the heuristic search approach for feature selection in unsupervised learning problems can be far superior than traditional baselines such as PCA and random projections.

Introduction

Clustering algorithms such as the k-means (McQueen 1967) group together objects that are more similar according to a distance metric in the objects' feature space. Often in real-world applications objects have a large number of features. For example, microarray gene expression datasets might contain several thousand features (Pomeroy et al. 2002). In such cases, clustering algorithms may suffer from the *curse of the dimensionality*. That is, due to the large number of features, it is hard to measure the distance of the objects in the feature space—the curse of dimensionality makes it hard for clustering methods to find meaningful groups of objects.

A common approach for preventing the curse of dimensionality is to select a discriminative subset of features as a clustering preprocessing step. If the objects have b features, then there is 2^b different subsets in the space of search, which is clearly too large for any reasonable b. One way of finding good discriminative subsets in such large search spaces is by employing a heuristic function to guide search algorithms to more promising feature subsets.

Hall (2000) introduced Correlation-Based Feature Selection (CFS), an algorithm which uses best-first search for feature subset selection for supervised learning. CFS uses a heuristic function which accounts for the feature-feature and feature-class correlations of a given subset to guide search. Gaudel and Sebag introduced FUSE, an algorithm that uses the UCT search algorithm (Kocsis and Szepesvári 2006) with the nearest-neighbor classifier as heuristic (reward) function for the UCT playouts (Gaudel and Sebag 2010). Both CFS and FUSE rely on the object labels to derive heuristic functions to guide search. In this paper we deal with problems where no labels are available.

We consider two settings of the clustering problem. In the first setting no background information is available. In the second setting, known as constrained-based clustering (CBC), in addition to the objects and their features, the user provides a set of pairwise *must-link* and *cannot-link* constraints to aid the process of clustering. Objects with a must-link constraint should be in the same cluster, while objects with a cannot-link constraint should not be in the same cluster (Wagstaff and Cardie 2000; Wagstaff et al. 2001).

We make the following three contributions in this paper. First, we introduce two heuristic functions based on must-link and cannot-link constraints to guide search. Second, we adapt an existing unsupervised quality metric for clustering to function as a heuristic to guide search. Finally, we present experiments testing different search algorithms guided by our heuristic functions in real-world datasets with tens, hundreds and thousands of features. Our results show that the heuristic search approach for feature selection in unsupervised learning problems can be far superior than traditional approaches such as PCA and random projections.

Related Work

Works from different communities such as machine learning, data mining and computer vision have dealt with the problem of feature selection. However, most of the feature selection algorithms have been developed and tested on supervised learning problems. In this section we review feature selection methods for unsupervised learning and methods that use state-space search for feature selection.

Yusta (2009) use the GRASP method for searching for discriminative subset of features for supervised learning problems. Similarly to the work of Hall (2000) and Gaudel and Sebag (2010), the algorithm of Yusta also relies on the class labels to derive a heuristic function to guide search. Procopiuc et al. (Procopiuc et al. 2002) present a Monte Carlo algorithm for feature selection in projective clustering. In projective clustering different clusters can be projected onto different subspaces. In this paper we search for a single discriminative subspace for all clusters—our problem is conceptually different from Procopiuc et al.'s. Fern and Brodley (Fern and Brodley 2003) present a feature selection

method for clustering problems based on ensembles of random projections—we use Fern and Brodley's method as a baseline comparison in our experiments.

Yip et al. (Yip, Cheung, and Ng 2005) proposes an objective function for handling feature selection during clustering. Li et al. (Li, Dong, and Ma 2008) use the information provided by pairwise constraints within the Expectation-Maximization algorithm to perform clustering and feature selection simultaneously. These two works are similar to ours in that they perform feature selection for clustering while using background knowledge (class labels in former and constraints in the latter). However, in contrast with the heuristic search approach we take in this paper, the approaches of Yip et al. and Li et al. do not tackle the whole combinatorial problem of feature selection.

To the best of our knowledge this is the first work presenting an empirical study of feature selection for unsupervised learning from a state-space search perspective.

Problem Formulation

Let $\mathbf{O}$ be a set of objects and $\mathbf{F}$ the set of features, with $|\mathbf{F}| = b$. Also, each object $o \in \mathbf{O}$ is associated with a class label $l(o)$.[1] Let $G = (V, E)$ be a directed graph describing the feature selection state space. Here, each state $n \in V$ represents one of the 2^b possible feature subsets. Also, for each state $n \in V$ $child(n) = \{n' | (n, n') \in E\}$ represents the child-parent relationship of feature subsets. Let F_n denote the set of features of state n, $child(n)$ has one child n' with features $F_{n'} = F_n \cup f$ for each feature $f \in \mathbf{F} \setminus F_n$.

In feature selection one is interested in finding $F \subseteq \mathbf{F}$ that maximizes the clustering quality of the partitioning a clustering algorithm is able to find. The start state is defined as the empty feature subset.

Heuristic Search Algorithms

In this section we briefly review the heuristic search algorithms we use in our study, which are Monte Carlo Random Walks (Nakhost and Müller 2009), Hill-Climbing, and GRASP (Feo and Resende 1995). All three algorithms start from the empty feature subset and iteratively add features to the subset until reaching the desired number of features. We now describe the iterative process of each algorithm.

Hill Climbing (HC) In each step HC evaluates all children of the current state and moves to the best child according to the employed heuristic function. The process is repeated until reaching a state in which no improvement is possible or when reaching the number of features selected by the user.

Monte Carlo Random Walks (MRW) In each step MRW performs M random walks of length L from the current state and uses the heuristic function to evaluate only the L-th state in each walk. After finishing all random walks MRW moves to the state which has the best heuristic value. This process is repeated until MRW reaches a state with the number of features equal to the number N determined by the user. This is the simplest version of MRW, which we use in our experiments — see (Nakhost and Müller 2009) for enhancements.

GRASP GRASP alternates greedy with local search. That is, first it performs a greedy search to find an initial solution and then it applies local search for improving such initial solution. In each iteration of the greedy search GRASP selects the top K children according to the heuristic function and randomly chooses one of them for expansion. This process is repeated until reaching a state s which contains the number of desired features. That is when GRASP performs a local search from s by randomly replacing features in s with those not in s. GRASP repeats the greedy search followed by the local search R times and returns the best state encountered according to the heuristic function.

Heuristic Functions for Feature Selection

Ideally one would use the metric used to measure clustering quality as heuristic to guide search. This way, finding a state s with a good heuristic value will likely help finding a good partition of the objects while using the subspace defined by s. However, in a typical clustering problem one does not have access to the function which measures partitioning quality during clustering nor during feature selection. We derive heuristic functions to operate as surrogates to the actual quality metric to guide search.

The first heuristic we present, the Silhouette metric, does not use background knowledge to guide the search. The other two heuristics, Constrained Objects Distance and Number of Satisfied Constraints use background knowledge in the form of must-link and cannot-link constraints.

The Silhouette Metric (SM)

The silhouette (Rousseeuw 1987) is a metric often used to evaluate clustering quality when class labels are absent. In this paper we use such metric to guide the search for discriminative feature subsets.

The silhouette of object $o \in \mathbf{O}$ is defined as,

$$SM(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}, \quad (1)$$

where $a(o)$ is the average distance between o and all objects in o's cluster. Intuitively, if o is in a cluster in which all objects are close to each other, then the value of $a(o)$ will be small. Let C be a cluster with $o \notin C$ and with the smallest average distance d between o and all objects in C (C is referred as the neighbor cluster of o). We have that $b(o) = d$. If $a(o)$ is much smaller than $b(o)$, then $SM(o)$ will be close to 1, which means that o is assigned to a good cluster. Similarly, if $SM(o)$ is close to -1, then there is a neighbor cluster C that perhaps o should have been assigned to. The SM is the average of the $SM(o)$-values for all $o \in \mathbf{O}$. Our search algorithms will search for a subset that maximizes the SM.

Constrained-Objects Distance (COD)

Let ML be the set of pairwise must-link constraints. That is, for objects o and o' with $(o, o') \in ML$, one knows that o and o' should be in the same cluster. Similarly we define CL

[1] In contrast with the supervised learning setting, if labels are available, we use them for verifying the clustering quality, not for guiding the process of finding F.

as the set of pairwise cannot-link constraints, and objects o and o' should not be in the same cluster if $(o, o') \in CL$. In this work we consider the ML and CL constraints as soft constraints. That is, the constraints provide useful information we can use to guide search, but they will not necessarily be satisfied in the partition of the objects our method encounters. Moreover, we use such constraints only for feature selection, not for clustering the objects.

The first heuristic function we present measures the distance in the feature space between constrained objects. We call this heuristic Constrained-Objects Distance (COD). Intuitively, a good subspace is the one in which a pair of objects in a must-link constraint are close to each other, and a pair of objects in a cannot-link constraint are far apart. The following equation captures this intuition,

$$COD(F, ML, CL) = \sum_{c,d \in CL} D_F(c, d) - \sum_{a,b \in ML} D_F(a, b)\,,$$

where $D_F(a, b)$ is the distance between objects a and b in the feature subspace F. Our objective is to find a subspace which maximizes the value of COD. By maximizing the COD-value we minimize the distance of objects in ML and maximize the distance of objects in CL.

Number of Satisfied Constraints (NSC)

In NSC, for a given set of objects $\mathbf{O}$, subspace F and sets ML and CL, one clusters the objects and counts the number of constraints which were satisfied. Our search algorithms will search for a subspace that maximizes NSC.

Empirical Evaluation

We evaluate HC, MRW, and GRASP using SM, COD, and NCS on the following datasets (the numbers in parenthesis show the number of objects and the number of features in each data set, respectively): Spectrometer (531, 93), Satellite (213, 36), Ionosphere (351, 34), Gissete (6000, 5000), and Arcene (100, 10000); see (Lichman 2013) for more details.

k-means is the clustering algorithm used after selecting the feature subset to partition the objects and also to compute the values of SM and NSC during search. Clustering quality is measured according to Rand Index (Rand 1971). The Rand Index measures the agreement of the clustering result with the class labels, where a value of 1 means perfect agreement between clustering and class labels. Euclidean distance is the metric used. We fix $|F| = 10$. We experimented with other values of $|F|$ and found results similar to the ones presented in this section.

Due to the stochasticity of the search algorithms employed in our experiments, we run 20 independent runs of each algorithm and report the average results. Standard deviations are omitted for space from the table of results because they were small. In addition to the average Rand Index ("Rand"), we also report the average running time of the algorithms in seconds ("Time"). The running time accounts for the time required to select the feature subset and to run k-means on the selected subspace. We use PCA and ensemble random projections ("Ensemble") as baseline comparisons. Note that we do not present results of PCA on Gissete and Arcene datasets because our PCA implementation is too slow to be practical in datasets with thousands features.

MRW and GRASP have input parameters which allow one to trade, to some extent, feature subset quality and running time. In MRW one can choose different number of random walks and in GRASP one can choose the number of times R the algorithm is run (restarts) and the number of children K considered in each step of the greedy search. The value of R is set to 10 and K is set to half of the children of any given node. In order to ensure a fair comparison of the algorithms, we choose MRW's number of random walks in a way that MRW's running time is similar to GRASP's.

The constraints are obtained in our experiments by randomly selecting pairs of objects o_1 and o_2 and if $l(o_1) \neq l(o_2)$, then a cannot-link constraint between o_1 and o_2 is added to CL; a must-link constraint is added to ML otherwise. We choose the number of constraints used in each experiment based on the number of objects in each dataset. That is, datasets with more objects allow more constraints. The number in parenthesis after COD and NSC shows the number of constraints used in each experiment.

The Rand Index values we present are computed over all objects in the dataset. We also computed the Rand Index values for the objects in the *held-out* dataset, which contains only objects not belonging to constraints. We observed that the Rand Index values when using all objects were similar to those when using the held-out dataset. Due to space constraints we present results only for the whole dataset.

Discussion

The results are presented in Table 1. We start by discussing the running time of the search algorithms. As expected, we observe that the search algorithms are substantially faster when using COD than when using SM or NSC. For example, HC takes 4,575.83 seconds on average to find a partition using SM in Spectrometer. By contrast, the same algorithm takes only 1.81 seconds on average when using COD. Similar results are observed with GRASP: in Satellite GRAPS takes 533.30 seconds on average to halt when using NSC (80) and only 34.55 seconds on average when using COD (80). COD is much faster than SM and NSC because COD does not run the k-means algorithm for each heuristic value computed—SM and NSC do.

The constrained-based heuristic functions perform either similarly or better than the unsupervised SM. This is because COD and NSC use the information provided by the constraints. As examples, in Satellite the best Rand Index obtained using SM is of 0.69 (MRW), while HC obtains an average Rand Index of 0.79 when using NSC (80). As another example, in Arcene, the best average Rand Index obtained while using SM is by MRW, which is of 0.36, while GRASP obtains an average of 0.44 with COD (40). All heuristics yielded similar results in Spectrometer, however.

In general, providing more constraints to COD and NSC helps improving the results as the heuristics become more informed. However, this is not a rule in Table 1. We notice a drop in the Rand Index value when increasing the number of constraints in a few cases. For example, see GRASP using COD (40) and COD (80) in Satellite. We conjecture that this

Spectrometer (531 objects and 93 features)										
Search Algorithm	SM		COD (50)		NSC (50)		COD (200)		NSC (200)	
	Rand	Time	Rand	Time	Rand	Time	Rand	Time	Rand	Time
HC	0.89	4,575.83	0.88	1.81	0.77	181.57	0.88	2.82	0.85	183.57
GRASP	0.80	44,921.30	0.88	145.79	0.79	2,120.82	0.88	627.92	0.79	2,118.54
MRW	0.89	80,294.10	0.89	46.96	0.71	3,011.44	0.89	184.53	0.76	3,191.15
PCA, Ensemble	0.53, 0.79									
Satellite (213 objects and 36 features)										
Search Algorithm	SM		COD (40)		NSC (40)		COD (80)		NSC (80)	
	Rand	Time	Rand	Time	Rand	Time	Rand	Time	Rand	Time
HC	0.60	208.82	0.76	0.26	0.74	12.53	0.76	0.44	0.79	12.34
GRASP	0.67	2,073.83	0.76	20.34	0.74	161.30	0.73	34.55	0.75	533.30
MRW	0.69	1,796.94	0.73	30.20	0.76	216.31	0.76	30.82	0.76	637.95
PCA, Ensemble	0.67, 0.72									
Ionosphere (351 objects and 34 features)										
Search Algorithm	SM		COD (50)		NSC (50)		COD (100)		NSC (100)	
	Rand	Time	Rand	Time	Rand	Time	Rand	Time	Rand	Time
HC	0.69	480.89	0.77	0.28	0.62	26.81	0.80	0.47	0.61	26.47
GRASP	0.60	4,621.73	0.64	22.38	0.67	266.65	0.79	36.48	0.63	322.54
MRW	0.59	4,961.82	0.79	37.86	0.63	300.46	0.64	38.44	0.55	304.79
PCA, Ensemble	0.72, 0.58									
Gisette (6,000 objects and 5,000 features)										
Search Algorithm	SM		COD (100)		NSC (100)		COD (500)		NSC (500)	
	Rand	Time	Rand	Time	Rand	Time	Rand	Time	Rand	Time
HC	0.49	9,423,555.00	0.70	180.05	0.65	68,302.00	0.71	431.76	0.63	83,916.00
GRASP	0.50	54,234,122.00	0.67	22,316.60	0.71	312,937.00	0.72	62,245.80	0.65	334,534.00
MRW	0.45	3,069,260.00	0.69	129.17	0.65	84,753.10	0.72	376.46	0.67	86,319.10
Ensemble	0.55									
Arcene (100 objects and 10,000 features)										
Search Algorithm	SM		COD (15)		NSC (15)		COD (40)		NSC (40)	
	Rand	Time	Rand	Time	Rand	Time	Rand	Time	Rand	Time
HC	0.28	16,219.39	0.39	314.54	0.36	2,289.21	0.37	285.34	0.40	2,213.90
GRASP	0.31	215,299.00	0.38	34,886.70	0.36	47,754.60	0.44	41,917.50	0.38	62,281.30
MRW	0.36	8,047.82	0.38	110.81	0.37	1,205.42	0.38	133.04	0.38	1,119.75
Ensemble	0.27									

Table 1: Clustering quality in terms of Rand Index and running time of search algorithms for feature selection.

happens due to possible misleading information provided by the constraints—we recall that the constraints are derived from the class labels which could be noisy.

Our table of results suggests there is no clear winner among the search algorithms—they all behave similarly in most of the problems. Our results suggest that as long as the search algorithm uses a good heuristic, then it will be able to find a good feature subset. There seems to be, however, a difference between the heuristic functions tested. In addition to being faster to compute, COD is able to provide better guidance than SM and NSC in a few cases; see for example the results for Gisette and Ionosphere.

Our last and arguably the most important observation is that search methods can be far superior than the baselines PCA and Ensemble of Random Projections. For example, in the Spectrometer domain, k-means using the subset of features selected by the search algorithms is able to find partitions with Rand Index of 0.88-0.89. By contrast, k-means finds partitions with Rand Index of 0.53 and 0.79 when using the subset selected by PCA and Ensemble, respectively. Ionosphere is the only domain in which the baselines are competitive or even superior in a few cases, but overall the heuristic search methods are far superior.

Concluding Remarks

In this paper we introduced two constrained-based heuristics, Constrained-Objects Distance (COD) and Number of Satisfied Constraints (NSC), and one unsupervised heuristic based on the Silhouette Metric (SM) for guiding the search for discriminative feature subsets in clustering problems. We also conducted an extensive experiment testing different search algorithms while using each of the three heuristics introduced. Our results showed that COD is much faster to compute and is often able to provide better guidance than NSC and SM. Moreover, our heuristic search approaches were far superior than baseline methods such as PCA and Ensemble of Random Projections in most of the cases tested.

Acknowledgement

This research was supported by CAPES, CNPq, and FAPEMIG.

References

Feo, T. A., and Resende, M. G. C. 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6:109–133.

Fern, X., and Brodley, C. 2003. Random projection for high dimensional data clustering: A cluster ensemble approach. *Proceedings of the International Conference on Machine Learning* 186–193.

Gaudel, R., and Sebag, M. 2010. Feature selection as a one-player game. In *Proceedings of the International Conference on Machine Learning*, 359–366.

Hall, M. A. 2000. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the International Conference on Machine Learning*, 359–366.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *Proceedings of the European Conference on Machine Learning*, 282–293. Springer-Verlag.

Li, Y.; Dong, M.; and Ma, Y. 2008. Feature selection for clustering with constraints using jensen-shannon divergence. In *Proceedings of the IEEE International Conference on Pattern Recognition*, 1–4.

Lichman, M. 2013. UCI machine learning repository.

McQueen, J. B. 1967. Some methods of classification and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Nakhost, H., and Müller, M. 2009. Monte-Carlo exploration for deterministic planning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1766–1771.

Pomeroy, S. L.; Tamayo, P.; Gaasenbeek, M.; Sturla, L. M.; Angelo, M.; McLaughlin, M. E.; Kim, J. Y. H.; Goumnerova, L. C.; Black, P. M.; Lau, C.; Allen, J. C.; Zagzag, D.; Olson, J. M.; Curran, T.; Wetmore, C.; Biegel, J. A.; Poggio, T.; Mukherjee, S.; Rifkin, R.; Califano, A.; Stolovitzky, G.; Louis, D. N.; Mesirov, J. P.; Lander, E. S.; and Golub, T. R. 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* 415(6870):436–442.

Procopiuc, C. M.; Jones, M.; Agarwal, P. K.; and Murali, T. M. 2002. A Monte Carlo algorithm for fast projective clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 418–427. ACM Press.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66:846 – 850.

Rousseeuw, P. J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20:53 – 65.

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

Wagstaff, K.; Cardie, C.; Rogers, S.; and Schroedl, S. 2001. Constrained K-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning*, 577–584.

Yip, K. Y.; Cheung, D. W.; and Ng, M. K. 2005. On discovery of extremely low-dimensional clusters using semi-supervised projected clustering. In *Proceedings of the IEEE International Conference on Data Engineering*, 329–340.

Yusta, S. C. 2009. Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30(5):525–534.