Feature Selection for Unsupervised Learning via Comparison of Distance Matrices

Stephan Dreiseitl

Dept. of Software Engineering
Upper Austria University of Applied Sciences
A-4232 Hagenberg, Austria

Abstract. Feature selection for unsupervised learning is generally harder than for supervised learning, because the former lacks the class information of the latter, and thus an obvious way by which to measure the quality of a feature subset. In this paper, we propose a new method based on representing data sets by their distance matrices, and judging feature combinations by how well the distance matrix using only these features resembles the distance matrix of the full data set. Using articial data for which the relevant features were known, we observed that the results depend on the data dimensionality, the fraction of relevant features, the overlap between clusters in the relevant feature subspaces, and how to measure the similarity of distance matrices. Our method consistently achieved higher than 80% detection rates of relevant features for a wide variety of experimental configurations.

Keywords: Unsupervised feature selection, feature extraction, dimensionality reduction, distance matrix similarity.

1 Introduction

In many machine learning applications, feature selection constitutes an important data preprocessing step. The expected benefits of feature selection are numerous, and can be broadly separated into two groups [1]: First, machine learning algorithms benefit from the removal of noise in the form of irrelevant and redundant features, as this helps to avoid overfitting and thus allows the models to generalize better. Second, in many application domains (such as biomedical informatics) feature selection is an important endeavor in its own right, helping to identify and highlight key aspects of the data. Often, this form of exploratory data analysis is the basis of subsequent research efforts in the application domain (e.g., which of a set of biomarker candidates is relevant for a biomedical problem).

While there is a substantial amount of literature on feature selection methods for supervised learning tasks [2–4], there is considerably less on methods for unsupervised learning. The reason for this discrepency lies in the fact that supervised problems provide a target value to predict, and it is easy to measure the performance of an algorithm on a feature set by how well it performs this

R. Moreno-Díaz et al. (Eds.): EUROCAST 2013, Part I, LNCS 8111, pp. 203–210, 2013.

prediction. Features that achieve high performance are more important than those that do not. Consequently, filter and wrapper methods were developed to identify features that work well, either individually or as feature sets.

Feature selection for unsupervised learning is hard because it lacks a clear-cut performance measure. Some approaches, collectively known as *subspace clustering* [5], find feature sets that cluster well, although different sets may lead to different clusters. Other methods implement wrappers around clustering algorithms such as k-means or the EM algorithm, and maximize criteria based on intra- vs. intercluster separation [6]. Entirely different approaches are to identify features that cluster well by the entropy of the distribution of all between-points distances [7, 8], or to cluster the features themselves by their linear dependency, and then picking representative features from each cluster [9].

This work proposes a different and new direction for feature selection in unsupervised learning tasks by using distance matrices to assess the relevance of features. The basis of this work is the observation that the clustering of a data set is entirely dependent on its spatial arrangement, which — for clustering purposes — can be represented by its distance matrix. The distance matrix of the entire data set is therefore the “gold standard” against which we can measure the quality of feature subsets.

We present the derivation and details of our new method in the next section, and demonstrate its efficacy in Sec. 3. A discussion of these results, along with concluding remarks, is given in Sec. 4.

2 Concepts and Methods

Our research hypothesis is that the distance matrix of a data set can be used to identify features in the data set that cluster well. The idea behind this hypothesis is as follows: Because the spatial arrangement of a data set can be represented by its distance matrix, relevant features are those for which the restricted distance matrix (using only these features) closely matches the original distance matrix. We should then be able to identify these relevant features by a simple greedy search algorithm. Although this algorithm provides a feature *ranking*, we can perform feature *selection* by choosing only the top-ranked features. In Sec. 2.4, we propose a method to automatically determine how many of these top-ranked features to choose.

2.1 Feature Ranking

Throughout this paper, we use X to denote the $n \times m$ data matrix, i.e., X contains n rows of data points of dimensionality m. For an index set $S \subseteq \{1, \ldots, m\}$, let X_{-S} be the data set with the columns (features) in S removed. Let $D(x)$ denote the distance matrix of a data set x, i.e., a symmetric, non-negative matrix with entry $D(x)_{ij}$ denoting the distance between the i^{th} and j^{th} entries in x.

A simple greedy backwards elimination algorithm for ranking features according to their relevance is given by following pseudocode:

```
1. Set $F \leftarrow \{1, \ldots, m\}$ to the set of all feature indices. Let $S \leftarrow \emptyset$ denote an initially empty set of feature indices. Calculate the full distance matrix $\tilde{D} \leftarrow D(X)$.
2. While $|F| > 1$ do:
   (a) For each $j \in F$, calculate $\tilde{D}_{-j} \leftarrow D(X_{-(S \cup \{j\})})$, the distance matrix with features $S \cup \{j\}$ removed.
   (b) Calculate $j^* \leftarrow \arg\min_{j \in F} \mathrm{sim}(\tilde{D}, \tilde{D}_{-j})$ as the feature for which the current distance matrix and the distance matrix with feature $j$ removed have the smallest similarity as calculated by a similarity measure sim.
   (c) Set $F \leftarrow F \setminus \{j^*\}$, $S \leftarrow S \cup \{j^*\}$, and recalculate $\tilde{D} \leftarrow D(X_{-S})$.
3. The reverse order in which feature indices are entered into $S$ gives an indication of their relevance.
```

In this algorithm, we recalculate the distance matrix for the remaining features after a feature is removed, rather than keeping one fixed distance matrix (the one for the entire data set). This is done to remove the contribution of features that are found to not be as relevant as the others.

2.2 Matrix Similarity Measures

In step 2(b) of the pseudocode above, we need to calculate the similarity between two distance matrices. There are a number of ways to accomplish this, most notably by

- the Pearson correlation coefficient ρ of the matrix entries;
- the R_V coefficient of matrix similarity [10], which for symmetric square matrices is given by

$$R_V(A, B) := \frac{\mathrm{tr}(A \cdot B)}{\sqrt{\mathrm{tr}(A \cdot A)\,\mathrm{tr}(B \cdot B)}}\,,$$

  with $\mathrm{tr}(A) = \sum_{i=1}^{n} A_{ii}$ denoting the trace of a square matrix;
- the symmetric version of the Kullback-Leibler divergence [11], used for measuring the distance between two probability distributions (for this, matrix entries first have to be normalized to sum 1):

$$\mathrm{KL}(A, B) := \sum_{j,k} A_{jk} \log \frac{A_{jk}}{B_{jk}} + \sum_{j,k} B_{jk} \log \frac{B_{jk}}{A_{jk}}\,.$$

  Because the Kullback-Leibler divergence is a distance — rather than a similarity — measure, we modified the greedy search described in Sec. 2.1 to discard the *most* distant feature when using this measure for assessing matrix similarity.

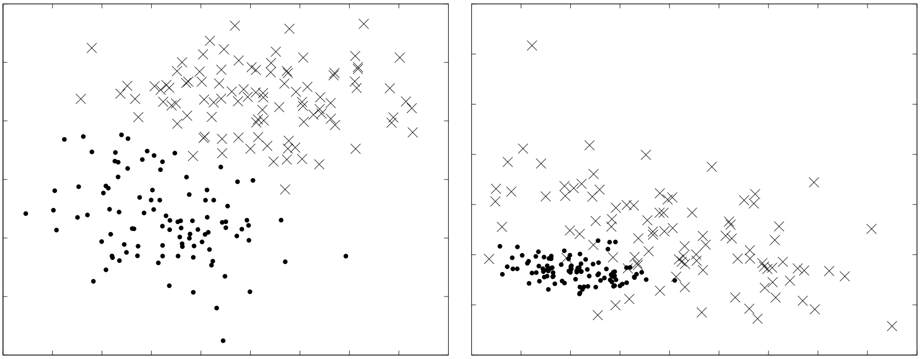

Fig. 1. Illustration of Bhattacharyya distance. Two features in five-dimensionals clusters with distance 50 (left), and two clusters with distance 10 (right).

2.3 Data Generation

In the experiments summarized in Sec. 3, we use artificial data so that we can control the data dimensionality and the ratio of relevant to irrelevant features. In particular, the relevant features comprise two multivariate Gaussians of varying overlap. The irrelevant features contain uniformly distributed noise. Both relevant and irrelevant features are subsequently scaled to zero mean and unit variance. All data sets contain 250 items in each of the two clusters.

In preliminary experiments, we observed that our algorithm is susceptible to the distance between the clusters in the data, with features more easily being recognized as relevant if the clusters are well separated. In order to quantify the contribution of this factor to our analyses, we constructed the relevant features as two multivariate Gaussians, for which the separation can be measured in the form of the *Bhattacharyya distance* [12, 13]

$$\mathrm{dist}(G_1, G_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\left(\frac{\det(\Sigma)}{\sqrt{\det(\Sigma_1)\det(\Sigma_2)}}\right).$$

Here, $G_1 = (\mu_1, \Sigma_1)$ and $G_2 = (\mu_2, \Sigma_2)$ are parametrizations of multivariate Gaussians, and $\Sigma = \frac{1}{2}(\Sigma_1 + \Sigma_2)$ is the arithmetic mean of the two covariance matrices. Fig. 1 provides an illustration of this notion of cluster distance.

2.4 Determining Relevant Features

When we plot the distance matrix similarity (or distance, in case of the Kullback-Leibler measure) vs. the removed features we can observe that there is a noticable difference between relevant and irrelevant features. Fig. 2 illustrates this observation for a 20-dimensional data set with 5 relevant features. There seems to be an exponential increase in the distance between distance matrices, as more and more features are removed from the data set. This progression no longer holds

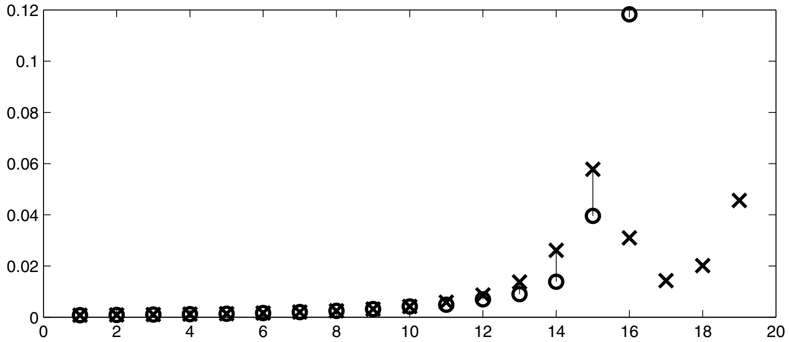

Fig. 2. Kullback-Leibler divergence between full distance matrix and distance matrix with the worst feature removed, shown as ×. The symbols o mark the predicted next value, assuming an exponential fit for the previous (left) values of ×. The feature at position 16 is the first that is deemed relevant by the heuristic of assessing the difference between × and o.

when the relevant features are reached at index 16 in the figure: An exponential fit to the previous values would predict a much larger distance than actually observed. Note that only 19 features are shown, because only that many features are removed from the original set of 20.

We therefore propose to select those features as relevant for which the exponential increase in similarity measure no longer holds. This process can be automated, with the first relevant feature being the one where there is more than 50% relative error between actual similarity measure and exponential model fit (calculated using the features up to this point). The value of 50% is a heuristic that seems to work reasonably well for various combionations of data dimensionality and number of relevant features.

3 Experiments

The experiments described here test the hypothesis that the greedy feature ranking algorithm in Sec. 2.1 is capable of identifying those features that belong to the artificially generated signal rather than to the noise. Within this broad setup, we investigate a number of experimental questions:

- Is one of the three methods for calculating the similarity of distance matrices (Pearson ρ, R_V coefficient, Kullback-Leibler divergence) better suited than the others?
- What effect does the data dimensionality, in particular the ratio of relevant to irrelevant features, have on the ability to detect relevant features?
- What influence does the distance between the clusters have on the results?

In the following, we will first rate combinations of these criteria by how well they can detect the relevant features, i.e., which percentage of the n truly relevant

Table 1. Percentage of n truly relevant features in the top n ranked features for each of three similarity measures and different combinations of relevant and total features. Table entries are averages over 100 runs. Bhattacharyya distance of clusters for all experiments was ~ 50.

	number of relevant features n (out of total data dimension)					
	1 (of 10)	2 (of 10)	3 (of 10)	5 (of 50)	10 (of 50)	15 (of 50)
similarity by:						
Pearson ρ	0	0.86	0.85	0.798	0.75	0.67
R_V coeffient	1	0.9	0.903	0.812	0.775	0.719
K-L divergence	1	0.935	0.953	0.83	0.835	0.792

features are in the top n features as ranked by the greedy search algorithm. We will then investigate the ability of the heuristic for automatically selecting relevant features that we described in Sec. 2.4. All experiments were repeated 100 times with different random numbers; the reported numbers are the averages over these 100 runs.

Table 1 gives a general impression of how well the three similarity/distance measures work in identifying the relevant features. One can observe that across all data dimensionalities and number of relevant features, the Kullback-Leibler divergence seems to be the best suited for measuring the difference between distance matrices in the greedy feature ranking algorithm. We therefore focus exclusively on this measure for the remainder of this paper. Furthermore, Table 1 also indicates that for constant data dimensionality (here 10 or 50), the percentage of truly relevant features among the highly-ranked features sometimes increase with the number of truly relevant features, and sometimes decreases. This is, for example, visible in the last row of this table, with the detection percentage rising from 0.935 to 0.953 for dimension 10, and falling from 0.835 to 0.792 for dimension 50.

To obtain a better understanding of the interaction between data dimensionality, number of relevant features, and cluster separation, we calculated the percentage of relevant features detected for all combinations of three data dimensions (10, 15, and 20), three number of relevant features (10%, 15%, and 20%), as well as seven different cluster overlaps (Bhattacharyya distances from 10 to 40), but now only for the Kullback-Leibler divergence as distance measure. The results of this comprehensive investigation is given in Table 2. As expected, increasing the cluster separation while keeping the dimensionality and number of relevant features constant (i.e., looking at each row in the table) generally results in higher numbers of detected relevant features, although there are a small number of fluctuations. Comparing entries in each 3×7 block of numbers for constant dimensionality with the corresponding entries in the other two such blocks, one can observe that the percentage of truly relevant features in the top ranked features mostly decreases with increasing data dimensionality. It thus becomes increasingly harder to identify relevant features in higher-dimensional spaces.

Table 2. Percentage of truly relevant features in the top ranked features for each of three data dimensions, in combination with different percentages of truly relevant features and Bhattacharyya distances. Table entries are averages over 100 runs.

	Bhattacharyya distance between clusters						
	10	15	20	25	30	35	40
dimensions = 10:							
relevant = 10%	0.99	1.00	1.00	1.00	1.00	1.00	1.00
relevant = 20%	0.62	0.69	0.70	0.68	0.81	0.74	0.69
relevant = 30%	0.74	0.80	0.79	0.78	0.81	0.80	0.79
dimensions = 20:							
relevant = 10%	0.84	0.84	0.91	0.89	0.90	0.91	0.92
relevant = 20%	0.82	0.86	0.85	0.89	0.84	0.87	0.85
relevant = 30%	0.81	0.83	0.85	0.87	0.86	0.90	0.91
dimensions = 30:							
relevant = 10%	0.78	0.82	0.82	0.86	0.90	0.89	0.90
relevant = 20%	0.71	0.77	0.81	0.83	0.83	0.86	0.84
relevant = 30%	0.69	0.74	0.79	0.83	0.86	0.88	0.88

Table 3. Percentage of times (out of 100 runs) that the heuristic of Sec. 2.4 was exactly correct, or off by at most one, when detecting the correct number of correct features in the data set. The percentage of relevant features was fixed to 30%.

	Bhattacharyya distance between clusters						
	10	15	20	25	30	35	40
exactly correct:							
dimensions = 10	0.56	0.54	0.57	0.48	0.53	0.53	0.61
dimensions = 20	0.32	0.36	0.43	0.49	0.45	0.48	0.51
dimensions = 30	0.05	0.16	0.21	0.26	0.35	0.36	0.24
approximately correct:							
dimensions = 10	1.00	1.00	1.00	1.00	1.00	1.00	1.00
dimensions = 20	0.69	0.73	0.81	0.85	0.82	0.83	0.82
dimensions = 30	0.22	0.24	0.48	0.57	0.77	0.77	0.75

The final part of our investigation consisted of checking how often the heuristic of Sec. 2.4 for determining the number of relevant features was correct. Since this may be too stringent a requirement, we checked both for the number of times where the heuristic was exactly correct, and for the number of times where it was off by at most one. We focused on a subset of the combinations in Table 2, looking only at the highest percentage of correct features (30%) for all three data dimensionalities. It can be seen that when settling for the approximate number of correct features, the heuristic is surprisingly accurate, with more than 75% correct for all data dimensionalities and large enough Bhattacharyya distances. The numbers for other percentages of relevant features are slightly lower (data not shown).

4 Conclusion

Without a clear-cut measure against which to assess a method's performance, unsupervised feature selection is generally harder than supervised feature selection, and less widely investigated. In this paper, we proposed a method for unsupervised feature selection that uses the distance matrix of a data set as a proxy for a gold standard against which to measure the performance of feature subsets. Although we implemented only a very crude greedy search mechanism for feature ranking, we nevertheless observed that relevant features can be distinguished from noise with high accuracy. More sophisticated search strategies, such as ones based on evoluationary computation, may lead to even better results.

References

1. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
2. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517 (2007)
3. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
4. Liu, H., Motoda, H., Setiono, R., Zhao, Z.: Feature selection: An ever evolving frontier in data mining. In: Proceedings of the 4th International Workshop on Feature Selection in Data Mining, pp. 4–13 (2010)
5. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensinal data: A review. ACM SIGKDD Explorations 6, 90–105 (2004)
6. Dy, J., Brodley, C.: Feature selection for unsupervised learning. Journal of Machine Learning Research 5, 845–889 (2004)
7. Dash, M., Liu, H.: Feature selection for clustering. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 110–121. Springer, Heidelberg (2000)
8. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature selection for clustering — a filter solution. In: Proceedings of the Second International Conference on Data Mining, pp. 115–122 (2002)
9. Mitra, P., Murthy, C., Pal, S.: Unsupervised feature selection using feature similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1–13 (2002)
10. Escoufier, Y.: Le traitement des variables vectorielles. Biometrics 29, 751–760 (1973)
11. Kullback, S., Leibler, R.: On information and sufficiency. Annals of Mathematical Statistics 22, 79–86 (1951)
12. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distribution. Bulletin of the Calcutta Mathematical Society 35, 99–109 (1943)
13. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, San Diego (1990)