Making Sense of Touch: Unsupervised Shapelet Learning in Bag-of-words Sense

Zhicong Xian, Tabish Chaudhary and Jürgen Bock

***Abstract*— A robot manipulation task often requires the correct perception of an interactive environment. Environment perception relies on either visual or haptic feedback. During a contact rich manipulation, vision is often occluded, while touch sensing proves to be more reliable. And a force sensor reading often entails abundant time series segments that reflect different manipulation events. These discriminating time series sub-sequences are also referred to as shapelets. The discovery of these shapelets can be considered as a clustering problem and the distance of sample time series to these shapelets is essentially a feature learning problem. Additionally, shapelets can also be considered as dictionaries in compressed sensing. This paper proposes a neural network with t-distributed stochastic neighborhood embedding as a hidden layer (NN-STNE) to project a long time series into a membership probability to a set of shorter time-series sub-sequences, i.e., shapelets. In this way, the dimensions of input data can be reduced. To preserve the local structure within data in the projected lower-dimensional space as in its original high dimensional space, a Gaussian kernel-based mean square error is used to guide the unsupervised learning. And due to the non-convex nature of the optimization problem for shapelet learning, K-means is used to find initial shapelet candidates. Different from existing shapelet/feature/dictionary learning algorithms, our method employs t-stochastic neighborhood embedding to overcome the crowding problem in projected low-dimensional space for shapelet learning. Moreover, our method can find an optimal length of the shapelets using L_1-norm regularization. The proposed method is then evaluated on the UCR time series dataset and an electrical component manipulation task, such as switching on, to prove its usefulness on improving clustering accuracy compared to other state-of-art feature learning algorithms in the robotic context.**

I. Introduction

In a contact-rich manipulation task, such as grasping, assembly, and handling, different errors may occur. Interpreting and understanding the touch sensor information can help us identify the root causes. For instance, a force impulse is detected in a time interval that is not supposed to be present. This may indicate a possible collision of a robot on an obstacle in an environment. Another example would be a time delay of a certain event represented by a force pattern, such as a level shift. This could suggest a change in the environment to interact with.

Uncovering and identifying certain force patterns can not only help us in diagnosing a robot application but also can be used for robot motion segmentation or detection of a schematic change in a robot application offline.

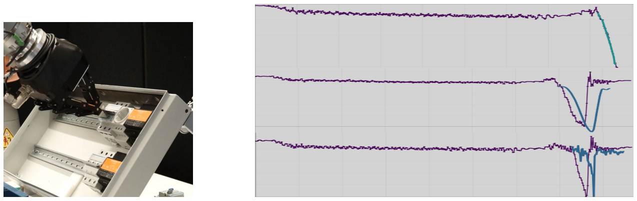

Fig. 1: A robot application: switch pushing task (left) and examples of identifying discriminative time series sub-sequences that distinguish a time series from others recorded in this application (right)

An use case application is illustrated in Fig.1, where a KUKA LBR iiwa 7kg tries to slide the lever up on the switch mounted on an electrical cabinet. Here two different errors can occur: the switch is already on, i.e., the lever of the switch is already pointing up; the switch is broken, where the lever can not be easily flipped on. The amplitude of external forces at the robot flange are depicted in Fig.1, where the purple curves are the min-max normalized trajectories of external forces at robot flange from three different classes, with two classes for unknown error and one for a normal case. The three additional plotted sub-sequences on the purple curves are learned shapelets, which highlight the differences between the trajectories from three different classes. In general, the extraction of time series sub-sequences, i.e., shapelets, can be termed as an unsupervised feature learning problem. Therefore, this paper aims at extracting discriminative features from the time-series data that can increase clustering accuracy.

In summary, the contribution of this paper is stated as follows:

- This paper presents a novel approach for unsupervised shapelet learning in bag-of-words sense.
- In essence, this approach (NN-STNE) is to learn interpretable features in an unsupervised way. It outputs similarity measures between an original time series and a list of shapelets. Using these transformed similarity measures as input features for clustering UCR open time-series dataset, we prove that this feature learning algorithm achieves competitive results compared to other state-of-art feature selection algorithms.

*This work was supported by KUKA Germany GmbH

Zhicong Xian, Jürgen Bock and Tabish Chaudhary are with Corporate Research, KUKA Germany GmbH, Zugspitzstrasse 140, 86165 Augsburg, {zhicong.xian, juergen.bock}@kuka.com

Workshop on New Advances in Brain-Inspired Perception, Interaction and Learning,
International Conference on Robotics and Automation 2020

II. Related Work

Shapelets [1] are originally proposed as time series subsequences that are considered as discriminating patterns for classification of temporal sequences. The basic idea to discover shapelets is to assess all possible segments from time-series data based on a merit function that measures the predictive power of a given sub-sequence for some class labels. In [1] at first a large set of shapelet candidates are generated and their similarity to time series segments across all the training samples is computed using the brute-force algorithm. Then a decision tree algorithm is applied to recursively split training samples into different subsets by selecting a shapelet candidate to maximize the information gain for classification.

Unsupervised feature selection entails shapelet learning, because a shapelet can also be considered as a time series feature. The challenge of feature selection without class labels is to discover uncorrelated and discriminative features. Different algorithms have been introduced. One of the state-of-art algorithms is using dictionary learning to discover shapelet in a generative way [2]. It considers a shapelet as an atom of dictionary and requires sliding the original time series into sub-sequences of the same length as shapelet, i.e., a dictionary atom. By trying to reconstruct slid sub-sequences, the algorithm jointly optimizes the shapelet dictionary and its corresponding sparse encoding for slid sub-sequences. This can result in losing global information of the original time-series and falling into the pitfalls of time series sub-sequence clustering [3]. Besides focusing on the shape feature of time series, each sample point in a time series can also be referred to as a potential feature similar to using pixels as input features for image clustering. [4] presents a $l_{2,1}$-norm regularized discriminative feature selection algorithm for unsupervised learning (UDFS) .

The presented algorithm in the scope of this paper differs from the above mentioned methods, in which we employ t-stochastic neighborhood embedding to perform a non-linear mapping between original data and shapelets and also attains the interpretability of the selected features.

III. Preliminaries

In this paper, scalar variables are denoted by unbold alphabets such as $(a, b, c, \alpha, \beta, \gamma, \cdots)$ whereas vector variables are denoted by bold lower-case alphabets $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{\alpha}, \cdots)$ and matrices by upper-case alphabets $(\boldsymbol{A}, \boldsymbol{B}, \cdots)$. The index of samples is represented by n; the index of a point in a time-series by q; the index of a point in a shapelet by m; the index of shapelets by k; the index for the class labels by c. Note that the unspecified parameters are denoted by lower-case letters while a constant defined number is represented with an upper-case letter, such as N, C, K, etc. For instance, the number of samples is denoted by N, the length of a time-series is written as Q, the length of patterns is represented with M, the number of patterns is K and the total number of class labels is denoted by C.

IV. Unsupervised Shapelet Learning with Stochastic Neighborhood Embedding (SNE)

In this section, we aim to introduce the novel unsupervised shapelet learning model with deep embedding. Since a shapelet is a discriminative time-series segment, its length is always shorter than the input time series. Therefore, a sliding window approach from [6], which slides the input time series into equal size of shapelet length, is presented at first. Then follows the calculation of similarity between the candidate shapelets and the time series sub-sequences. After that, for each candidate shapelet, its corresponding most similar time series sub-subsequence from one input time series is selected. The normalized cross-correlation score for measuring the similarity between training data sub-subsequence and candidate shapelet is then converted into a probability distribution. The more similar they are, the higher the probability to assign the time series sub-sequence to this candidate shapelet will be. Based on this probability distribution, the corresponding shapelet candidates will be updated to an extent that is proportional to its assignment probability. Fig.2 illustrates the whole architecture at a glance.

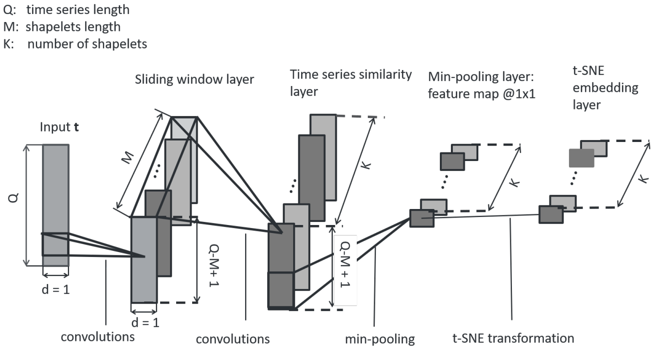

Fig. 2: Overview of the network architecture

A. Layers in NN-STNE Network

Initial Estimation of Number and Length of Shapelets For the length of shapelets we adopt the same strategy as in [6]. By human inspection, we estimate a possible length of shapelets, e.g., M. The number of shapelets, i.e., K , of size M, is chosen in a way, such that a large number of input time series, i.e., N, of length Q from different classes C can be represented using only K number of shapelets. It can be interpreted as a vector quantification problem with each shapelet as one bit. Then it is equivalent to solve the following equation:

$$2^K = N \times (Q - M) \times C \qquad (1)$$

By solving the above equation, one can obtain $K = \log_2[N \times (Q - M) \times C]$ assuming that each shapelet is completely different from each other.

Sliding Window Approach Given a one-dimensional time-series sample, i.e. $\boldsymbol{t} = [t_1, t_2, t_3, \cdots, t_q]$ of length Q, and a desired shapelet to learn as $\boldsymbol{s} = [s_1, s_2, \cdots, s_m]$ of length M. In order to perform a point-to-point similarity measurement calculation, it is required that the segments to be compared should be of equal size. Since the expected length of a shapelet is often pre-defined according to some heuristics, the input time-series can be split into segments of equal length as shapelets. To achieve this, the implementation of [6] is used. The sliding window approach is realized by the convolution operation. And therefore, a time series of length Q can be divided into $J := Q - M + 1$ sub-sequences of equal size of M.

Time Series Similarity Layer After sliding the input time-series into equal size, each will then be compared to a shapelet candidate. There are many different metrics for measuring time-series similarity: cross-correlation [7], dynamic time warping [8], Euclidean distance. In the scope of this paper, we adopt the normalized cross-correlation for measuring time series similarity. In this way, the obtained shapelets are not simply the average of matching time series sub-sequences from the input, but rather only a segment in a shapelet that better correlates with another segment in a slid sub-sequence from input time series get updated. Consequently, the characteristics in the input time series, such as, sharp changes, can be captured better. The calculation of cross normalized correlation in [7] is applied and we use fast Fourier transform (FFT) to speed up the computation [9]. By [7], the normalized cross correlation can be expressed as:

$$NCC(\boldsymbol{s}_k, \boldsymbol{t}_{i,j}) = \max_{\omega} \left(CC_{\omega}(\boldsymbol{s}_k^z, \boldsymbol{t}_{i,j}^z) \right) \tag{2}$$

$$D_{i,j,k} = 1 - NCC(\boldsymbol{s}_k, \boldsymbol{t}_{i,j}) \tag{3}$$

with $\boldsymbol{s}_k^z$, $\boldsymbol{t}_{i,j}^z$ as z-normalized k-th shapelet of length M and j-th slid window of length M from the i-th time series sample respectively. And ω is the amount of right or left shift of a time series sub-sequence according to the definition of cross correlation, which is expressed as a function $CC_{\omega}(\cdot)$. Additionally, $NCC(\boldsymbol{s}_k, \boldsymbol{t}_{i,j})$ is a function that denotes the normalized cross correlation between k-th shapelet $\boldsymbol{s}_k$ and time series sub-sequence $\boldsymbol{t}_{i,j}$ and $D_{i,j,k}$ is an entry in the time series similarity matrix $\boldsymbol{D} \in \mathbb{R}^{N \times J \times K}$. By the definition of normalized cross correlation, it has a range between -1 and 1, i.e., $NCC(\boldsymbol{s}_k, \boldsymbol{t}_{i,j}) \in [-1, 1]$. For simplicity and avoid negative activation in neural network, we subtract the normalized cross correlation score from 1 to obtain $D_{i,j,k}$ as defined in (3). The smaller the $D_{i,j,k}$, the more similar the sub-sequence and a shapelet candidate will be.

T-Distributed Stochastic Neighborhood Embedding (t-SNE) Layer

After transforming the time series data, i.e., $\boldsymbol{T} \in \mathbb{R}^{N \times Q}$ into distances to different shapelets, i.e., $\boldsymbol{D} \in \mathbb{R}^{N \times J \times K}$ Then follows the selection of for each shapelet most matching sub-sequence in one time series sample by min-pooling, i.e., $F_{i,k} = \min_j \boldsymbol{D}$ with $D_{i,j,k}$ as an entry in $\boldsymbol{D}$ and $F_{i,k}$ as an entry from the new matrix, i.e., $\boldsymbol{F} \in \mathbb{R}^{N \times K}$.Therefore, we can represent a time series sample, i.e., $\boldsymbol{t}_i \in \mathbb{R}^Q$ by the distances to different shapelets, i.e., $\boldsymbol{f} \in \mathbb{R}^K$ as i-th row in $\boldsymbol{F} \in \mathbb{R}^{N \times K}$, where we reduce the information amount from original Q, i.e. the length of time-series, into K, i.e. the number of shapelets. Here a shapelet can be considered as a coordinate axis in lower-dimensional map space.

When a high-dimensional data is mapped into a low-dimensional space, a crowding problem could occur [10]. Inspired by [11], a t-student distribution is employed to convert the similarity between a most matching sub-sequence and the shapelet candidates into probability:

$$q_{i,k} = \frac{(1 + F_{i,k})/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_k \left(1 + F_{i,k}/\alpha)^{-\frac{\alpha+1}{2}}\right)} \tag{4}$$

where α is the degree of freedom of student t-distribution and $F_{i,k}$ is a distance metric always larger than 0, and the smaller the distance metric, the higher the similarity score. In the following experiment, we let $\alpha = 1$ [12].

B. Objective Function

Spectral Analysis. It is assumed that two similar time series should share similar distances to candidate shapelets.To describe this, a mean square error scaled by Gaussian kernel from the spectral analysis is adopted [13]. Consider that $\boldsymbol{G} \in \mathbb{R}^{N \times N}$ is the matrix for describing similarity among all the time series samples with an entry defined as: $G_{(ij)} = \mathrm{e}^{-\frac{\left\|\boldsymbol{t}_i - \boldsymbol{t}_j\right\|^2}{\sigma^2}}$ with σ denotes the variance of the Gaussian kernel and $\boldsymbol{t}_i$, $\boldsymbol{t}_j$ are for two different time series samples. The variance of the Gaussian kernel also defines the effective number of neighbors for a given sample point [14]. With this, the Gaussian kernel based mean square error is expressed as:

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{(ij)} \left\| \boldsymbol{q}_{(i,:)} - \boldsymbol{q}_{(j,:)} \right\|_2^2 \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{(ij)} \left[q_{(k,i)} - q_{(k,j)} \right]^2 \\
&= \sum_{k=1}^{K} \boldsymbol{q}_{(k,:)}^T \left(\boldsymbol{D}_G - G \right) \boldsymbol{q}_{(k,:)} \\
&= tr\left(\boldsymbol{q}^T \boldsymbol{L}_G \boldsymbol{q} \right)
\end{aligned} \tag{5}$$

where $\boldsymbol{q}_{(i,:)}, \boldsymbol{q}_{(j,:)} \in \mathbb{R}^{1 \times K}$ are the transformed distances of time series sample i, j to shapelet candidates respectively and N is the number of time series samples. In addition, $\boldsymbol{L}_G = \boldsymbol{D}_G - \boldsymbol{G}$ is the Laplacian matrix in spectral analysis with $\boldsymbol{D}_G$ as a diagonal matrix with element defined as $D_G(i,i) = \sum_{j=1}^{n} G_{(ij)}$ Besides this, it is also important that different shapelets should be as distinct to each other as possible.

Encouraging diverse shapelets. Again we employ Gaussian kernel to penalize similar shapelets [13]. The similarity between shapelets can be described as $\boldsymbol{H} \in \mathbb{R}^{K \times K}$, where an entry is defined as $H_{(i,j)} = \mathrm{e}^{-\frac{||\boldsymbol{s}_i - \boldsymbol{s}_j||^2}{\sigma^2}}$. And hence, to encourage distinct shapelet is to minimize the norm of shapelet similarity matrix, i.e., $||\boldsymbol{H}||_2^2$.

Automatic selection of shapelets length. Shapelets are the convolutional kernel weights in the time series similarity layer as shown in Fig.2. To find out the optimal length of shapelet, it is equivalent to applying regularization techniques on kernel weights, where a L_1 norm is introduced to force shapelet filters to become zeros if possible. As a result, the zeros in shapelets will not contribute to the calculation of normalized cross-correlation. Consequently, the true length of shapelets can be obtained by removing zero values in the shapelet values.

In summary, the total objective function to minimize is formulated as:

$$\mathcal{L} = tr\left(\boldsymbol{q}^T \boldsymbol{L}_G \boldsymbol{q}\right) + \lambda ||\boldsymbol{H}||_2^2 + \beta \sum_{k=1}^{K} \sum_{l=1}^{M} |s_{k,l}| \qquad (6)$$

with λ as a weighting factor of the minimization of shapelet similarity and the last term $\beta \sum_{k=1}^{K} \sum_{l=1}^{M} |s_{k,l}|$ as a shapelet regularization term to automatic select optimal length of shapelets. β is a parameter given by the user to weigh the trade-off between learning of more similar shapelets to the input data and generalization.

V. Experiment and Evaluation

In this section, the proposed method unsupervised shapelet learning with t-distributed stochastic neighborhood embedding layer is tested on public time-series dataset and its performance is evaluated compared to other unsupervised feature and shapelet learning techniques as mentioned in Sec. II such as Uncorrelated and Discriminative Feature Selection (UDFS) [4], k-Shape [7] and unsupervised shapelet learning [13].

A. Data Sets

To make our evaluation comparable to other state-of-art shapelet learning algorithms, a subset from the public open data sets UCR [1] is used. On the other hand, we also need to prove its usefulness in the robotics application. Therefore, the data from the KUKA LBR iiwa switching on application as depicted in Fig.1 is used. And a description of the used data-sets is presented in Table I

B. Evaluation Metrics

To evaluate the performance of clustering, different evaluation metrics, such as Accuracy(ACC), Normalized Mutual Information(NMI) [15], can be applied. To have a comparative study on the method proposed in [13], we also use the Rand Index to evaluate our algorithm.

TABLE I: Statistics of benchmark time series data set.

Data set	Train/Test	Length	# Classes
ECG 200	100/100 (200)	96	2
CBF	30/900 (930)	128	3
Face Four	24/88 (112)	350	4
OSU Leaf	200/242(442)	427	6
Robot Switch Push up	489/81(570)	433	3

TABLE II: Comparison of different algorithms in terms of clustering performance

Data set	KMeans	UDFS +KMeans	NN-STNE +KMeans
ECG 200	0.6	0.55	**0.7**
CBF	0.74	0.73	**0.93**
Face Four	0.74	0.73	**0.81**
OSU Leaf	**0.76**	0.76	0.76
Switching Up	0.74	0.74	**1.0**
Average	0.72	0.7	**0.84**

C. Comparison Results

Since our algorithm transforms time series data into distances to shapelets and does not directly output pseudo-labels, to prove its usefulness, we consider the transformed shapelet distances as extracted features and feed them to a clustering method, such as KMeans. Consequently, a predicted label can be obtained. Then we compare our feature learning algorithms NN-TSNE with other feature learning algorithms mentioned in Sec.II such as, UDFS [16], using Rand Index defined in [13]. We select the best results obtained by UDFS using different number of neighborhood and list the results in Table II. The best result for each data set is highlighted in bold.

From Table II we can observe that in most of the cases using NN-STNE as feature selection algorithms before applying KMeans can help us achieve better clustering results than without applying any feature selection algorithms. Interesting is also to note that using UDFS as feature selection can somehow make the clustering result slightly worse. And from these five data sets, we can observe a 16.7% of improvement on clustering results using KMeans on average.

D. Conclusion

Time series data can be analyzed from either the temporal perspective or shape perspective. In this paper, we focused on the shape perspective of time series data and proposed NN-STNE as a feature learning algorithm to discover discriminative time-series sub-sequences using embedded learning and proved that, when shape features in time series data prevail, applying it as a feature selection step can help improve clustering accuracy.

[1] http://timeseriesclassification.com/dataset.php

References

[1] L. Ye and E. Keogh, “Time series shapelets: A new primitive for data mining,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. New York, NY, USA: ACM, 2009, pp. 947–956. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557122

[2] J. Zhang, X. Li, L. Gao, L. Wen, and G. Liu, “A shapelet dictionary learning algorithm for time series classification,” in *15th IEEE International Conference on Automation Science and Engineering, CASE 2019, Vancouver, BC, Canada, August 22-26, 2019*. IEEE, 2019, pp. 299–304. [Online]. Available: https://doi.org/10.1109/COASE.2019.8843231

[3] E. Keogh and J. Lin, “Clustering of time series subsequences is meaningless: Implications for past and future research,” in *In Proc. of the 3rd IEEE International Conference on Data Mining*, 2003, pp. 115–122.

[4] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, “l2, 1-norm regularized discriminative feature selection for unsupervised learning,” in *IJCAI*, 2011.

[5] M. Madry, L. Bo, D. Kragic, and D. Fox, “St-hmp: Unsupervised spatio-temporal feature learning for tactile data,” 05 2014, pp. 2262–2269.

[6] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, “Learning time-series shapelets,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’14. New York, NY, USA: ACM, 2014, pp. 392–401. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623613

[7] J. Paparrizos and L. Gravano, “k-shape: Efficient and accurate clustering of time series,” *SIGMOD Rec.*, vol. 45, no. 1, pp. 69–76, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2949741.2949758

[8] S. Salvador and P. Chan, “Fastdtw: Toward accurate dynamic time warping in linear time and space,” in *KDD workshop on mining temporal and sequential data*. Citeseer, 2004.

[9] J. Lewis, “Fast template matching,” *Vis. Interface*, vol. 95, 11 1994.

[10] G. E. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 833–840.

[11] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[12] L. van der Maaten, “Learning a parametric embedding by preserving local structure.” *Journal of Machine Learning Research - Proceedings Track*, vol. 5, pp. 384–391, 01 2009.

[13] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang, “Unsupervised feature learning from time series,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI’16. AAAI Press, 2016, pp. 2322–2328. [Online]. Available: http://dl.acm.org/citation.cfm?id=3060832.3060946

[14] L. Zelnik-manor and P. Perona, “Self-tuning spectral clustering,” in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 1601–1608. [Online]. Available: http://papers.nips.cc/paper/2619-self-tuning-spectral-clustering.pdf

[15] A. Strehl and J. Ghosh, “Cluster ensembles – a knowledge reuse framework for combining multiple partitions,” *Journal on Machine Learning Research (JMLR)*, vol. 3, pp. 583–617, December 2002. [Online]. Available: http://strehl.com/download/strehl-jmlr02.pdf

[16] X. Li, H. Zhang, R. Zhang, and F. Nie, “Discriminative and uncorrelated feature selection with constrained spectral analysis in unsupervised learning,” *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, 10 2019.