Euler Clustering

Jian-Sheng Wu, Wei-Shi Zheng*, Jian-Huang Lai

School of Information Science and Technology, Sun Yat-sen University, Guangzhou, P.R. China.
Guangdong Province Key Laboratory of Computational Science, Guangzhou, P.R. China.
jiansheng4211@gmail.com, *wszheng@ieee.org (corresponding author), stsljh@mail.sysu.edu.cn.

Abstract

By always mapping data from lower dimensional space into higher or even infinite dimensional space, kernel k-means is able to organize data into groups when data of different clusters are not linearly separable. However, kernel k-means incurs the large scale computation due to the representation theorem, i.e. keeping an extremely large kernel matrix in memory when using popular Gaussian and spatial pyramid matching kernels, which largely limits its use for processing large scale data. Also, existing kernel clustering can be overfitted by outliers as well. In this paper, we introduce an Euler clustering, which can not only maintain the benefit of nonlinear modeling using kernel function but also significantly solve the large scale computational problem in kernel-based clustering. This is realized by incorporating Euler kernel. Euler kernel is relying on a nonlinear and robust cosine metric that is less sensitive to outliers. More important it intrinsically induces an empirical map which maps data onto a complex space of the same dimension. Euler clustering takes these advantages to measure the similarity between data in a robust way without increasing the dimensionality of data, and thus solves the large scale problem in kernel k-means. We evaluate Euler clustering and show its superiority against related methods on five publicly available datasets.

1 Introduction

Clustering plays an important role in organizing, understanding and learning data. As the information science and Internet techniques develop, a large amount of data, especially image data, are created every day. Consequently, it challenges existing clustering models in both the computational aspect due to increasing size of data and the robustness aspect due to more diversities of data. It is particularly a big obstacle for applying kernel k-means as well as other related kernel based clustering methods [Schölkopf *et al.*, 1998; Ng *et al.*, 2001; Dhillon *et al.*, 2004] to analyze large scale data.

Most if not all, the obstacle of using kernel-based clustering methods for large scale data is mainly caused by the use of kernel trick. Kernel based clustering implicitly transforms data from lower dimensional space to higher or even infinite dimensional feature space (i.e., Reproducing Kernel Hilbert Space (RKHS)), and then linear clustering models are applied in that feature space. The mapping is implicitly induced and supposed to deal with nonlinear clustering problem.

However, relying on commonly used kernel functions, such as Gaussian kernel [Chitta *et al.*, 2011], spatial pyramid matching kernel [Lazebnik *et al.*, 2006] and etc., kernel based clustering methods could not be applied to large scale data. It is because, those frequently used kernel functions implicitly but not explicitly induce a mapping by using kernel trick[Schölkopf, 2000], and this makes the space and computational complexity of kernel based clustering increase dramatically due to computing and keeping the full kernel matrix during learning. Hence, existing kernel based methods are not appropriate for large scale clustering problems due to the limitation of memory, e.g. it needs about 7,500GB memory to keep the kernel matrix for the dataset with 1 million data points. Although the kernel matrix can be subdivided and loaded in the memory block by block [Zhang and Rudnicky, 2002], the optimization will cost the time that is quadratic to the size of the dataset, so that it does not work efficiently for large scale problems. To address the high computational complexity problem in the optimization, an approximate kernel k-means approach is proposed in [Chitta *et al.*, 2011].

In this work, we solve the large scale computation problem in kernel based clustering methods by incorporating Euler kernel [Liwicki *et al.*, 2012]. Particularly, we develop an Euler clustering, which is scalable kernel k-means. The Euler kernel is relying on a nonlinear and robust cosine metric, and more important it intrinsically induces an empirical map which maps data to a complex space of the same dimension. In addition, the cosine metric is a robust metric and is able to tolerate outliers during clustering. Euler clustering takes the advantages of Euler kernel to measure the similarity between data in a robust way without using kernel trick and increasing the dimensionality of data, and thus solves the large scale problem in kernel k-means.

Euler clustering differs existing clustering methods in the following aspects:

- Euler clustering gets rid of kernel trick and can directly calculate the corresponding distance between a data point and any cluster center efficiently, and thus no need to calculate and store the kernel matrix.

- The dimension of feature space corresponding to the mapping induced by Euler kernel is the same as data's dimensionality itself. This is because although Euler kernel is based on nonlinear cosine distance, it is however equivalent to deriving an explicit mapping, which maps data onto a complex space of the same dimension. Note that, since the mapping induced by Euler kernel has the same dimension as input data, the computational complexity of Euler kernel is comparable to k-means. Although there exists empirical kernel mapping methods [Schölkopf *et al.*, 1998; Ng *et al.*, 2001; Dhillon *et al.*, 2004; Chitta *et al.*, 2011], they either lose some discriminant information or lead to high dimensionality for each data after the mapping. Hence, Euler clustering is suitable for large scale clustering problems without deriving any approximate models.

- Less sensitive to noise and outliers. In Euler clustering, the distance between data points in the RKHS space is a cosine based measure. Due to the periodicity and the range of the cosine function, the affects of the noise features on the distance are lessened. Hence, outliers will pull the corresponding mean vectors used as the cluster representative prototypes to themselves much more lightly in the RKHS space.

Although Euler clustering is just a special case of kernel k-means, the contribution of this work lies in that by using Euler kernel, Euler clustering can address the large scale and robust clustering simultaneously in a simple but elegant and effective way. In addition, Euler kernel is very different from existing widely known Mercer kernels in that it is defined in the complex domain rather than the real domain. However, it can be used for distance based modeling and our work is able to tell the usefulness of complex methods for clustering analysis.

The reminder of this paper is organized as follows. Section 2 briefly introduces the kernel k-means algorithm and the Euler kernel. In section 3, we introduce Euler clustering and detail the fast optimization process. The used datasets and the experimental results are reported in section 4. Then we conclude this paper in section 5.

2 Preliminary

In order to make our paper self-contained, we first introduce kernel k-means and the Euler kernel which is defined in the complex domain.

2.1 Kernel k-means

Kernel k-means first maps data points from the data space into a RKHS space using an implicit mapping induced by kernel function through kernel trick, and then clusters data to find cluster assignments such that the similar data points are in the same cluster and the dissimilar data points are in different clusters.

More specifically, let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the given dataset consisting of n data points in $\mathcal{R}^d$, and let $\psi(\cdot)$ denote the mapping induced by kernel function, i.e. $\psi(\mathbf{x}_j)$ is the image of $\mathbf{x}_j$ in this defined RKHS space. To group the dataset $\mathcal{X}$

into C clusters, kernel k-means is modeled to minimize the distortion error [Bishop, 2006]

$$f = \sum_{c=1}^{C} \sum_{j=1}^{n} U_{jc} \|\psi(\mathbf{x}_j) - \mathbf{m}_c\|^2. \quad (1)$$

Here $\mathbf{m}_c$ is the representative prototype, which is given by the mean vector of the images of the cluster $\mathcal{G}_c$ as

$$\mathbf{m}_c = \frac{\sum_{\mathbf{x}_j \in \mathcal{G}_c} \psi(\mathbf{x}_j)}{\sum_{\mathbf{x}_j \in \mathcal{G}_c} U_{jc}}, \quad (2)$$

where U is the cluster membership indicator matrix, $U_{jc} = 1$ if $\mathbf{x}_j$ belongs to $\mathcal{G}_c$, and $U_{jc} = 0$ otherwise.

After obtaining the optimal representative prototypes $\mathbf{m}_c$, the cluster labels for data points are assigned by finding the closest cluster prototypes as follows

$$v_j \leftarrow \arg \min_{c=1,\ldots,C} \|\psi(\mathbf{x}_j) - \mathbf{m}_c\|^2, \quad (3)$$

where v_j is the assignment for data point $\mathbf{x}_j$.

However, due to the implicit representation of images in the RKHS space, the cluster prototypes cannot be represented explicitly, so that the distance between image $\psi(\mathbf{x}_j)$ and the c-th cluster prototype $\mathbf{m}_c$ must be calculated by using the kernel trick as follows

$$K_{j,j} - \frac{2 \sum_{\mathbf{x}_q \in \mathcal{G}_c} K_{j,q}}{n_c} + \frac{\sum_{\mathbf{x}_q \in \mathcal{G}_c} \sum_{\mathbf{x}_l \in \mathcal{G}_c} K_{q,l}}{n_c^2}, \quad (4)$$

where K is the corresponding kernel matrix defined by $\psi(\cdot)$, and n_c is the number of data points belonging to the c-th cluster. Obviously, it has to store the full kernel matrix, which may be too large to be loaded in the memory, and it has to spend $O(Cn)$ time in assigning label to each data point at each iteration according to Eq. (4). Hence in total $O(Cn^2)$ time is used to get labels assigned for data points at each iteration. Therefore, kernel k-means, by using traditional kernel, is not suitable for solving the clustering problem for large amount of data points.

2.2 Euler Kernel

Let $\mathbf{x}_j$ be one of data points in the given dataset $\mathcal{X}$. Denote the image set as $\Phi = \{\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)\}$, then the kernel matrix K is defined by $\phi(\cdot)$ as $K = \Phi^H \Phi$ [Paulsen, 2009]. That is, the (j, q)-entry of the kernel matrix is given as follows

$$\begin{aligned} K_{jq} &= \frac{1}{2} \sum_{c=1}^{d} \cos\left(\alpha\pi\left(\mathbf{x}_j(c) - \mathbf{x}_q(c)\right)\right) \\ &\quad - i\frac{1}{2} \sum_{c=1}^{d} \sin\left(\alpha\pi\left(\mathbf{x}_j(c) - \mathbf{x}_q(c)\right)\right). \quad (5) \end{aligned}$$

Different from existing Mercer kernels, K is defined in the complex space. And, we have $K^H = K$ because $K_{qj} = \overline{K}_{jq}$, where $\bar{\cdot}$ represents the complex conjugate operator.

It is found that Euler kernel actually induces a mapping $\phi(\mathbf{x}_j)$ [Liwicki *et al.*, 2012] that maps data point $\mathbf{x}_j$ from

d-dimensional real space $\mathcal{R}^d$ into d-dimensional complex RKHS space $\mathcal{C}^d$, i.e. $\phi : \mathcal{R}^d \to \mathcal{C}^d$, as follows

$$\phi(\mathbf{x}_j) = \frac{1}{\sqrt{2}}e^{i\alpha\pi\mathbf{x}_j} = \frac{1}{\sqrt{2}}(\cos(\alpha\pi\mathbf{x}_j) + i\sin(\alpha\pi\mathbf{x}_j)), \quad (6)$$

where i is the imaginary unit.

Therefore, the distance function $d(\cdot, \cdot)$ which calculates the square Euclidean distance between images $\phi(\mathbf{x}_j)$ and $\phi(\mathbf{x}_q)$ in the RKHS space is given by

$$
\begin{aligned}
d(\phi(\mathbf{x}_j), \phi(\mathbf{x}_q)) &= \|\phi(\mathbf{x}_j) - \phi(\mathbf{x}_q)\|^2 \\
&= \sum_{c=1}^{d}\left(1 - \cos\left(\theta_j(c) - \theta_q(c)\right)\right), (7)
\end{aligned}
$$

where θ_j denotes $\alpha\pi\mathbf{x}_j$. Hence, the distance function $d(\cdot, \cdot)$ is a real value, so that the distances can be used to measure the dissimilarities between data points, although the kernel function ϕ and the kernel matrix K are defined in the complex space.

Since cosine function is a periodic function and its value lies in the range $[-1, 1]$, the distance defined as Eq. (7) is less sensitive to noisy features than ℓ_2-norm (i.e. Euclidean form) defined in the data space by limiting the affects of noisy features on the distance in the range $[-1, 1]$. When the value α is small, the distance function Eq. (7) will become l_2-norm. As the value α increases, Eq. (7) will reduce the large distances caused by the noise features [Liwicki *et al.*, 2012].

By Eq. (6), Euler kernel can induce an explicit map from input data space to RKHS space without increasing the data dimension. Although Euler kernel function and the corresponding kernel matrix are defined in the complex domain, the distance between data points is ultimately formulated in the real domain, so that it can be used as the dissimilarity measure between data points.

3 Euler Clustering

In this section, we will propose a novel kernel based k-means approach, called Euler clustering. We will employ the Euler kernel function to define the RKHS space and organize the data into C clusters based on a k-means style algorithm in this defined RKHS space. Although it is a kernel based k-means approach, it is almost as fast as k-means algorithm and less sensitive to noise and outliers.

Since, Euler kernel can induce an explicit mapping $\phi(\cdot)$ from data space to feature space, the images in the RKHS space can be explicitly represented through Eq. (6) in the complex space. Hence, the cluster representative prototype $\mathbf{m}_c$ can be explicitly computed in the complex space by the following lemma.

Lemma 1. *Denote the size of the cluster $\mathcal{G}_c$ by n_c, then the optimal $\mathbf{m}_c$ can be represented explicitly as*

$$\mathbf{m}_c = \frac{1}{\sqrt{2}}(\mathbf{a} + i\mathbf{b}), \quad (8)$$

where $\mathbf{a}$ and $\mathbf{b}$ are given by $\mathbf{a} = \frac{1}{n_c}\sum_{\mathbf{x}_j \in \mathcal{G}_c}\cos(\alpha\pi\mathbf{x}_j)$ and $\mathbf{b} = \frac{1}{n_c}\sum_{\mathbf{x}_j \in \mathcal{G}_c}\sin(\alpha\pi\mathbf{x}_j)$.

The lemma is straightforward. Note that by conducting k-means algorithm in the RKHS space, the optimal $\mathbf{m}_c$ is given as the mean vector of the cluster $\mathcal{G}_c$, i.e.

$$\mathbf{m}_c = \frac{1}{n_c}\sum_{\mathbf{x}_j \in \mathcal{G}_c}\phi(\mathbf{x}_j). \quad (9)$$

Due to the explicit formulation of the mapping in Eq. (6) for Euler kernel function, we have

$$\mathbf{m}_c = \frac{1}{\sqrt{2}n_c}\left(\sum_{\mathbf{x}_j \in \mathcal{G}_c}\cos(\alpha\pi\mathbf{x}_j) + i\sum_{\mathbf{x}_j \in \mathcal{G}_c}\sin(\alpha\pi\mathbf{x}_j)\right). \quad (10)$$

This suggests that different from existing kernel k-means using Gaussian, spatial pyramid matching kernels and etc., the cluster centers $\mathbf{m}_c$ can be represented explicitly in Euler clustering. Based on this fact, we can derive the below criterion of Euler clustering.

Theorem 1. *Euler clustering is to minimize the distortion error*

$$f = \sum_{c=1}^{C}\sum_{j=1}^{n}U_{jc}d(\phi(\mathbf{x}_j), \mathbf{m}_c), \quad (11)$$

where $\mathbf{m}_c$ is given by Eq. (10), and the squared Euclidean distance $d(\mathbf{x}_j, \mathbf{m}_c)$ (defined in Eq. (7)) from the data point $\mathbf{x}_j$ to the cluster prototype $\mathbf{m}_c$ is given by

$$\frac{d}{2} + \|\mathbf{m}_c\|^2 - \cos(\alpha\pi\mathbf{x}_j)^T\mathbf{a} - \sin(\alpha\pi\mathbf{x}_j)^T\mathbf{b}. \quad (12)$$

Proof. Denote $\alpha\pi\mathbf{x}_j$ by θ_j and represent $\mathbf{m}_c$ explicitly as Eq. (10). The squared Euclidean distance $d(\mathbf{x}_j, \mathbf{m}_c)$ can be expanded as follows

$$
\begin{aligned}
d(\mathbf{x}_j, \mathbf{m}_c) &= \|\frac{\cos(\theta_j) - \mathbf{a}}{\sqrt{2}} + i\frac{\sin(\theta_j) - \mathbf{b}}{\sqrt{2}}\|^2 \\
&= \|\frac{\cos(\theta_j) - \mathbf{a}}{\sqrt{2}}\|^2 + \|\frac{\sin(\theta_j) - \mathbf{b}}{\sqrt{2}}\|^2 \\
&= \frac{d}{2} + \|\mathbf{m}_c\|^2 - \cos(\theta_j)^T\mathbf{a} - \sin(\theta_j)^T\mathbf{b}.
\end{aligned}
$$

So that the distance from $\mathbf{x}_j$ to $\mathbf{m}_c$ in the RKHS space can be calculated more efficiently. □

Euler clustering is in line with k-means. Euler clustering performs the following steps iteratively for optimizing the criterion: 1) updating the optimal cluster prototypes $\mathbf{m}_c$ explicitly using Eq. (10) by fixing the cluster labels of data points, and 2) grouping data points into closest clusters according to Eq. (12) by fixing $\mathbf{m}_c$. Algorithm 1 details the procedure. Euler clustering will monotonically decrease the objective function after updating the cluster representative prototypes and re-assigning labels for data points. Like k-means, it can be concluded that the objective function of Euler clustering is bounded as well. Therefore, the Euler clustering algorithm will converge to a local minimum in a finite number of iterations.

What more important is that the computational complexity of Euler clustering is comparable to k-means, although it is

---
Algorithm 1 Euler Clustering

1: **Input:** Dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the number of clusters C, the value α, and the maximum number of iterations t_{max}.
2: **Output:** The cluster sets $\mathcal{G}_c$, $c = 1, \ldots, C$.
3: **Initialization:**
   Initialize the cluster labels of data points.
   Transform the data points from data space onto feature space using Eq. (6).
4: **repeat**
5:    t = t + 1;
6:    Fix the cluster labels of the data points, and update the cluster representative prototypes $\mathbf{m}_c$ using Eq. (10).
7:    Fix the cluster representative prototypes $\mathbf{m}_c$, and update the cluster labels of the data points using Eq. (12).
8: **until** convergence $\|\ t > t_{max}$.
---

a nonlinear clustering method. More specifically, Euler clustering only takes $O(nd)$ and $O(ndC)$ to update the cluster representative prototypes and the cluster labels of data points in each iteration. the Euler clustering algorithm converges in $O(TndC)$, where T is the number of iterations. Hence, Euler clustering can code with the nonlinear clustering and perform as light as a linear method. The following theorem summarize the computational analysis of Euler Clustering.

Theorem 2. *The proposed Euler clustering algorithm can converge to a local optimum in $O(TndC)$, where T is the number of iterations.*

4 Experiments

In this section, we evaluate the proposed Euler clustering method on five public datasets: Event [Li and Fei-Fei, 2007], 13 Natural Scene Categories [Fei-Fei and Perona, 2005], Caltech 101 [Fei-Fei *et al.*, 2006], Imagenet [Deng *et al.*, 2009], and Caltech 256 [Griffin *et al.*, 2007], to show its performance compared to k-means algorithm [MacQueen, 1967], kernel k-means algorithm [Schölkopf *et al.*, 1998], normalized cut algorithm [Shi and Malik, 2000], and approximate kernel k-means algorithm [Chitta *et al.*, 2011]. The Gaussian kernel is a very effective kernel and commonly used in many fields, and the spatial pyramid matching kernel is often applied in computer vision. Both two kernels are applied in kernel k-means, normalized cut, and approximate kernel k-means. We denote them as G-K-Kmeans (P-K-Kmeans), G-N-Cut (P-N-Cut), and G-AppK-Kmeans (P-AppK-Kmeans) when using Gaussian kernel (spatial pyramid matching kernel), respectively.

4.1 Experimental Settings

For all the five datasets, the spatial pyramid image representation was created for each image. We first extracted 128-dimensional SIFT descriptors [Lowe, 1999] for each image. Then we employed the bag-of-features [Fei-Fei and Perona, 2005] to get 512 prototypes for the SIFT features and apply the spatial pyramid modeling [Lazebnik *et al.*, 2006] to obtain three-level based pyramid histogram representation, which is

a sparse 10,752-dimensional vector for representing each image.

We measure the clustering performance in terms of the normalized mutual information (NMI) [Dhillon *et al.*, 2004]. The NMI lies in [0, 1]. The larger the NMI is, the better the clustering matches the true class distribution. Meanwhile, the empirical time the compared approaches take to achieve the convergence is measured to show the efficiency of our method.

For all the experiments shown in the following, the value α of our approach is set to be 700. For the Gaussian kernel, the kernel width is set as $\sqrt{l \frac{\sum_{i=1}^n \sum_{j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\|^2}{n(n-1)}}$, where l is selected from the sequences $[0.001, 0.005, 0.01, \ldots, 100, 200, \ldots 1000]$. For the approximate kernel k-means algorithms, the sample size is set to be 1,000, such that its clustering performance on all the five datasets approximate the corresponding kernel k-means algorithm.

All the methods are implemented in Matlab and run on an Intel Xeon 2.67GHz processor. To make the performance of the compared methods comparable, all the results reported are averaged over 10 runs with 10 different initializations of the cluster labels of data points. All compared methods are conducted under the same setting. The maximum number of iterations is set to be 100 for all compared methods.

4.2 Datasets

Dataset	Size	Dimension	#Class
Event	1,579	10,752	8
Scene 13	3,859	10,752	13
Caltech 101	8,677	10,752	101
SImagenet	19,911	10,752	12
Caltech 256	29,780	10,752	256

Table 1: Five public datasets

Table 1 summarizes the properties of the five datasets, and the details of these datasets are shown as follows.

- Event: The event dataset contains 8 sports event categories collected from Internet with the sizes of categories varying from 137 to 250.

- 13 Natural Scenes: There are 3,859 images covering 13 categories of natural scenes in the 13 Natural Scenes dataset. The average size of each image is about 250×300 pixels.

- Caltech 101: This dataset consists of 8,677 pictures of objects belonging to 101 categories. There are about 40 to 800 images per category, and most categories have about 50 images. The images in this dataset are left-right aligned.

- SImagenet: The full Imagenet dataset contains 1,261,402 images that are organized into 1,000 leaf synsets in a synset tree, in which each leaf synset represents a class of images. We selected 19,911 images from 12 synsets: manhole cover, daily, website, odometer,

	Event	Scene 13	Caltech 101	SImagenet	Caltech 256
Kmeans	0.1326 ± 0.0306	0.255 ± 0.037	0.3227 ± 0.0082	0.2142 ± 0.0075	0.2206 ± 0.0042
G-K-Kmeans	0.227 ± 0.0084	0.3125 ± 0.0109	0.3344 ± 0.0088	0.2594 ± 0.0081	0.2342 ± 0.0051
P-K-Kmeans	0.3576 ± 0.0124	0.514 ± 0.0355	0.4475 ± 0.0051	0.4694 ± 0.0123	0.2965 ± 0.004
G-N-Cut	0.1357	0.2717	0.3535	0.2847	0.2775
P-N-Cut	0.3059	0.393	0.4458	0.4267	0.3054
G-AppK-Kmeans	0.2266 ± 0.0138	0.306 ± 0.0065	0.3365 ± 0.0093	0.26 ± 0.0088	0.2313 ± 0.0019
P-AppK-Kmeans	0.3584 ± 0.0143	0.5094 ± 0.0358	0.4447 ± 0.0066	0.4651 ± 0.0133	0.2943 ± 0.004
Euler-Clustering	**0.382 ± 0.0187**	**0.582 ± 0.0103**	**0.4869 ± 0.0066**	**0.5146 ± 0.0074**	**0.3175 ± 0.002**

Table 2: Average NMI over 10 runs on the five datasets

	Event	Scene 13	Caltech 101	SImagenet	Caltech 256
Kmeans	5.18 ± 1.35	21.03 ± 8.32	241.03 ± 47.46	166.86 ± 41.83	2141.69 ± 35.93
G-K-Kmeans	**4.09 ± 0.11**	24.26 ± 1.86	139.84 ± 9.26	638.44 ± 48.5	3871.81 ± 353.70
P-K-Kmeans	232.12 ± 0.15	244.15 ± 1.28	1412.48 ± 13.69	11522.27 ± 77.0	23292.58 ± 283.37
G-N-Cut	4.26	22.47	145.65	549.41	2682.05
P-N-Cut	195.17	240.80	1248.48	12140.52	20596.08
G-AppK-Kmeans	9.24 ± 0.51	**18.12 ± 1.06**	**61.09 ± 8.72**	**94.25 ± 6.7**	**345.3 ± 8.88**
P-AppK-Kmeans	228.97 ± 0.28	129.99 ± 1.39	311.91 ± 4.63	1207.53 ± 5.49	1545.59 ± 30.85
Euler-Clustering	8.23 ± 2.56	36.16 ± 9.94	156.53 ± 41.28	283.77 ± 101.59	2036.28 ± 297.73

Table 3: Average cost in seconds over 10 runs on the five datasets

monarch butterfly, rapeseed, cliff dwelling, mountain, geyser, shoji, door, villa, to construct a small dataset SImagenet [Chitta *et al.*, 2011].

- Caltech 256: The collection of this dataset contains 30,607 images of objects belonging to 256 categories and 1 clutter. We discarded the images in the clutter, and chose the other images to construct this dataset consisting of 29,780 images covering 256 categories. Compared to Caltech 101, there are more images per category, and the images are not left-aligned in this dataset, thus it is harder to identify.

4.3 Experimental Results

As aforementioned, we compare all methods in terms of the NMI and the computational cost on the five datasets in the Table 2 and Table 3. Table 2 tabulates the average NMI over 10 runs for the five public datasets. On all datasets, Euler clustering achieves the best clustering performance among all the compared approaches. It improves the clustering significantly.

Compared to Non-approximate kernel methods, such as k-means, Euler clustering enhances the clustering performance with almost 1~2 times higher average NMI on Event, Scene 13 and SImagenet, and 50% higher average NMI on Caltech 101 and Caltech 256. Compared to the Gaussian kernel based kernel k-means and normalized cut, Euler clustering obtains great improvement as well. In particular on Scene 13 dataset, Euler clustering obtains 0.27 higher average NMI against G-K-Kmeans, and 0.31 higher average NMI against G-N-Cut; on SImagenet dataset, 0.255 higher average NMI and 0.23 higher average NMI are achieved as compared to G-K-Kmeans and G-N-Cut respectively. Even compared to the spatial pyramid matching kernel based ones that are popular-

ly used in literature for image categorization, Euler clustering improves the clustering with almost 10%~15% higher NMI values as well.

Compared to approximate kernel clustering methods such as G-AppK-Kmeans and P-AppK-Kmeans, Euler clustering also gains significant improvement on the NMI performance. The figures in Table 2 and Table 3 show the approximate methods, especially G-AppK-Kmeans, is always much faster than Euler clustering. However the price is that its performance is still much lower than Euler clustering, although it may suggest a combination of the approximation methodology and Euler clustering would be a promising extension in future.

Among all datasets, Euler clustering performs much better on Scene 13, Caltech 101 and SImagenet. Although dataset Event contains less images and less categories, the clustering performances of all approaches are lower than the ones on Scene 13, SImagenet, and Caltech 101, because the images from Event contain more semantic information. All approaches do not work very well on Caltech 256 as shown in Table 2, since it is large and contains diverse categories with large intra- and inter- class variations. Nevertheless, Euler clustering still obtains comparable performance on the two datasets.

Regarding the stability of Euler clustering, Table 2 also shows that Euler clustering not only obtains the best clustering performance in terms of NMI but also obtains the most robust clustering performance in terms of the standard deviation of NMI values against initializations. It obtains the smallest standard deviation of NMI values on SImagenet and Caltech 256. Although the absolute standard deviation values on Scene 13 and Caltech 101 it obtains are not the smallest, the relative standard deviation values with respective to the
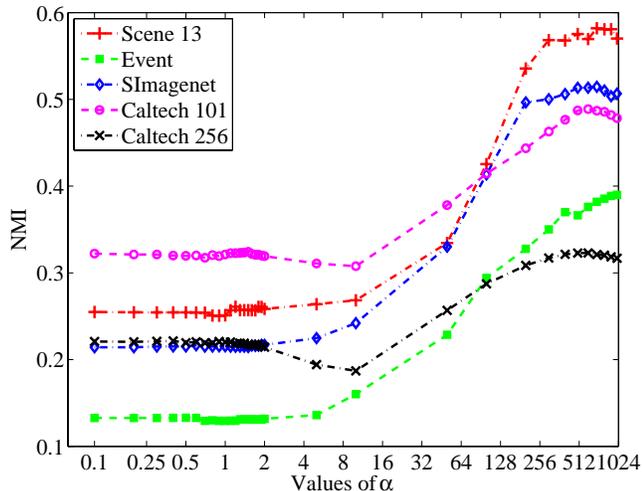
Figure 1: The clustering performance of Euler clustering on the five datasets with different α values

NMI values are the smallest. On Event, the relative standard deviation value it obtains is not the smallest, but it is not larger than the ones obtained by G-K-Kmeans and P-AppK-Kmeans too much.

Finally, the time for each method to take to achieve the convergence is shown in Table 3. Table 3 shows that Euler clustering is an efficient and more robust approach, although it is not the fastest on the five datasets. Compared to k-means, Euler clustering takes comparable time, as Euler clustering needs additional processing to compute the mapping. However, it converges with less iterations on the datasets Caltech 101 and Caltech 256. The reason why Gaussian kernel based clustering, namely G-K-Kmeans runs faster than Euler clustering and even k-means on Event and Scene 13 as shown in Table 1 is because the sizes of Event and Scene 13 are much smaller than the dimensionality of data, so the Gaussian k-ernel matrix can be calculated efficiently. However, as long as the size of dataset increases, G-K-Kmeans takes more and more time than Euler clustering on SImagenet and Caltech 256. This is because by the representation theorem, the computational complexity of G-K-Kmeans is a quadratic function of the number of samples and G-K-Kmeans therefore becomes more costly for large scale dataset. P-K-Kmeans always takes a lot of time, because modeling spatial pyramid takes time. Although P-K-Kmeans always gets better performance than G-K-Kmeans, it is however still clearly performs inferior to Euler clustering in the aspect of both computational cost and NMI performance (10%-15% lower NMI values than Euler clustering).

4.4 Parameters

In this section, we will show that our proposal is robust to the parameter value. For investigation, we plot the average NMI values it achieves on all the five datasets by varying α in the range $[0.1, 1000]$ in Figure 1.

Figure 1 shows that when α lies in the range $[0.1, 2]$, Euler clustering performs similarly to the k-means algorithm with

the NMI values being close to the ones achieved by k-means, which are shown in Table 2, because the cosine based distance ensembles the l_2 norm distance when α is small. As α increases, the clustering becomes better matching to the true class distribution. When α is in the range $[100, 1000]$, Euler clustering performs much better than k-means as shown in Figure 1. Figure 1 also shows that Euler clustering performs stably when α in the range $[100, 1000]$.

5   Conclusion

In this paper, we introduce Euler clustering in order to solve the existing largely unsolved problems for kernel clustering, namely the large scale computation problem and robust clustering. This is achieved by incorporating Euler kernel in line with kernel k-means. Euler kernel differs from existing kernels and is a complex function. It provides a cosine distance based measure between data and meanwhile induces an empirical mapping of the same dimension as input data. Our experiments have validated our approach against related approaches.

Acknowledgments

References

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Chitta *et al.*, 2011] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain. Approximate kernel k-means: Solution to large scale kernel clustering. In *proc. of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 895–903, 2011.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[Dhillon *et al.*, 2004] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means, spectral clustering and normalized cuts. In *proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 551–556, 2004.

[Fei-Fei and Perona, 2005] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[Fei-Fei *et al.*, 2006] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[Griffin *et al.*, 2007] Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[Lazebnik *et al.*, 2006] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[Li and Fei-Fei, 2007] Li-Jia Li and Li Fei-Fei. What, where and who? classifying events by scene and object recognition. In *proc. of IEEE 11th Int. Conf. on Computer Vision*, pages 1–8, 2007.

[Liwicki *et al.*, 2012] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Euler principal component analysis. *International Journal of Computer Vision*, 2012.

[Lowe, 1999] David G. Lowe. Object recognition from local scale-invariant features. In *proc. of IEEE 7th Int. Conf. on Computer Vision*, pages 1150–1157, 1999.

[MacQueen, 1967] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *proc. 5th Berkely Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[Ng *et al.*, 2001] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 2, 2001.

[Paulsen, 2009] Paulsen. An introduction to the theory of reproducing kernel hilbert spaces, 2009.

[Schölkopf *et al.*, 1998] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[Schölkopf, 2000] Bernhard Schölkopf. The kernel trick for distances. In *in Advances in Neural Information Processing systems*, pages 301–307, 2000.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Zhang and Rudnicky, 2002] Rong Zhang and Alexander I. Rudnicky. A large scale clustering scheme for kernel k-means. In *proc. of the 16th Int. Conf. on Pattern Recognition*, pages 289–292, 2002.