SOFT GEODESIC KERNEL K-MEANS

Jaehwan Kim [†], *Kwang-Hyun Shim* [†], *Seungjin Choi* [§]

[†] Digital Content Research Division, ETRI, Korea
{jh.kim,shimkh}@etri.re.kr
[§] Department of Computer Science, POSTECH, Korea
seungjin@postech.ac.kr

ABSTRACT

In this paper we present a kernel method for data clustering, where the soft k-means is carried out in a feature space, instead of input data space, leading to soft kernel k-means. We also incorporate a geodesic kernel into the soft kernel k-means, in order to take the data manifold structure into account. The method is referred to as soft geodesic kernel k-means. In contrast to k-means, our method is able to identify clusters that are not linearly separable. In addition, soft responsibilities as well as geodesic kernel, improve the clustering performance, compared to kernel k-means. Numerical experiments with toy data sets and real-world data sets (UCI and document clustering), confirm the useful behavior of the proposed method.

Index Terms**—** Pattern clustering methods, pattern classification, unsupervised learning

1. INTRODUCTION

Clustering, the goal of which is to partition data points into K coherent groups, plays an important role in a variety of areas such as pattern recognition, machine learning, data mining, computer vision, computational biology, and so on [1, 2]. k-means is one of widely-used clustering methods, where coherent clusters are identified in such a way that the sum of within-cluster variations (in terms of Euclidean distances) is minimized. Alternatively, it can also be interpreted as a minimization of the sum of squared pairwise intra-cluster distances [3]. A major limitation of k-means is that it cannot identify clusters which are not linearly separable in input space.

Kernel method[4] is a technique to tackle nonlinearly separable problems in an easy way, where the inner product between nonlinearly-transformed variables is replaced by an appropriate positive definite kernel (Mercer kernel) such that classification or clustering is carried out implicitly in a feature space. Kernel k-means was recently proposed [5] as an extension of the standard k-means algorithm, in order to overcome the limitation of k-means (mentioned above). It was shown in [5] that kernel k-means is closely related to spectral clustering and normalized cut.

Soft k-means is a slight variation of k-means, where responsibilities (degree to which a data point $\boldsymbol{x}_t$ is assigned to cluster k) are determined in a soft manner through the softmax function, whereas the standard k-means algorithm employs the hard decision (i.e., responsibilities are either 0 or 1). Although the minimum of the k-means objective function will be achieved with the responsibilities which are all either 0 or 1, the soft k-means with stiffness term which has an associated length scale improves the clustering performance, compared to the standard k-means algorithm [6].

In this paper we present a kernelized version of soft k-means, referred to as 'soft kernel k-means', which is derived in the same way as in kernel k-means. We also introduce a *geodesic kernel* which well reflects the data manifold structure. We incorporate the geodesic kernel into the soft kernel k-means algorithm, leading to our proposed clustering algorithm, soft geodesic kernel k-means. We show that the soft geodesic kernel k-means algorithm improves the clustering performance, compared to kernel k-means as well as soft kernel k-means with RBF kernel.

2. SOFT GEODESIC KERNEL K-MEANS ALGORITHM

We begin with revisiting the soft k-means algorithm. Then, we illustrate how to kernelize it as well as how to incorporate the geodesic kernel into the algorithm.

2.1. Soft k-means

The soft k-means method complements one of the main weak points of the standard (hard) k-means: data points assigned to a cluster have exactly the same degree of assignment without considering the distance between each data point and the mean of the cluster [6].

Given a set of data points $\{\boldsymbol{x}_t\}_{t=1}^N$, the soft k-means algorithm aims at identifying K disjoint clusters, $\{\mathcal{S}_j\}_{j=1}^K$, by iterating the following two steps [6]:

- **Assignment step:** Compute responsibility, $R_{jt} \in [0, 1]$,

$$R_{jt} = \frac{\exp\left\{-\beta\|\boldsymbol{x}_t - \boldsymbol{\mu}_j\|^2\right\}}{\sum_{i=1}^K \exp\left\{-\beta\|\boldsymbol{x}_t - \boldsymbol{\mu}_i\|^2\right\}}, \tag{1}$$

  that is the degree to which $\boldsymbol{x}_t$ is assigned to cluster $\mathcal{S}_j$, whose mean vector is denoted by $\boldsymbol{\mu}_j$. The assignment step has one parameter β that is referred to as the *stiffness*. The responsibility R_{jt} ranges over [0,1], whereas it is either 0 or 1 in the standard (hard) k-means. Note that the sum of the K responsibilities for each data point $\boldsymbol{x}_t$ is 1, i.e, $\sum_{j=1}^K R_{jt} = 1$ for $t = 1, \ldots, N$.
- **Update step:** The model parameters, means vectors, are adjusted to match the sample means of the data points that they are responsible for, i.e.,

$$\boldsymbol{\mu}_j = \frac{\sum_{t=1}^N R_{jt}\boldsymbol{x}_t}{\sum_{t=1}^N R_{jt}} = \frac{\sum_{t=1}^N R_{jt}\boldsymbol{x}_t}{\gamma_j}. \tag{2}$$

The soft k-means algorithm can be derived from the minimization of the following Lagrangian:

$$\mathcal{J}_{SK} = \underbrace{\sum_{t=1}^N \sum_{j=1}^K R_{jt}\|\boldsymbol{x}_t - \boldsymbol{\mu}_j\|^2}_{\text{expected energy}} - \underbrace{\frac{1}{\beta}\sum_{t=1}^N \sum_{j=1}^K R_{jt} \log \frac{1}{R_{jt}}}_{\text{entropy}}. \tag{3}$$

where $1/\beta$ is the Lagrange multiplier. As the stiffness parameter β approaches ∞, the entropy term disappears. In such a case, the Lagrangian (3) becomes identical to the cost function for the standard k-means. The entropy term encourages the soft assignments by spreading the responsibility of each data point uniformly as much as possible.

2.2. Geodesic kernel

We consider a nonlinear transform $\phi(\boldsymbol{x}_t)$ which is a mapping from input space to a feature space. We define a kernel matrix $\boldsymbol{K} = [K_{ij}] \in \mathbb{R}^{N \times N}$ where $K_{ij} = \phi^\top(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)$. Then, in the feature space, the squared Euclidean distance from $\phi(\boldsymbol{x}_t)$ to the best representative mean vector $\boldsymbol{\mu}_j$, is determined by the kernel without explicit knowledge of $\phi(\boldsymbol{x}_t)$ (*kernel trick*), i.e.,

$$\begin{aligned} \left\|\phi(\boldsymbol{x}_t) - \boldsymbol{\mu}_j\right\|^2 &= \left\|\phi(\boldsymbol{x}_t) - \frac{\sum_{l=1}^{N} R_{jl}\,\phi(\boldsymbol{x}_l)}{\sum_{l=1}^{N} R_{jl}}\right\|^2 \\ &= K_{tt} - \frac{2}{\gamma_j}\sum_{l=1}^{N} R_{jl}K_{tl} \\ &\quad + \frac{1}{\gamma_j^2}\sum_{n=1}^{N}\sum_{l=1}^{N} R_{jn}R_{jl}K_{nl}, \end{aligned} \tag{4}$$

where $\gamma_j = \sum_{l=1}^{N} R_{jl}$. This relation provides a central trick in developing the soft kernel k-means, which was also investigated in the weighted kernel k-means [5]. The basic idea of the soft kernel k-means algorithm is to evaluate responsibilities using the relation (4) through a kernel matrix $\boldsymbol{K}$.

Fig. 1 illustrates the (Dijkstra) geodesic distance on a neighborhood graph. The geodesic distance well reflects the local structure, which is expected to improve the clustering performance. Such a geodesic distance was employed in a low-dimensional embedding problem, which seeks an embedding with preserving neighborhood relations. Isomap is one exemplary method [**?**, **?**], which was further elaborated in the framework of kernel methods [7, 8].

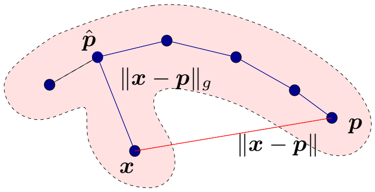

$\hat{\boldsymbol{p}}$
$\|\boldsymbol{x} - \boldsymbol{p}\|_g$
$\boldsymbol{p}$
$\|\boldsymbol{x} - \boldsymbol{p}\|$
$\boldsymbol{x}$

Fig. 1. Euclidean distance between nodes x and p, $\|\boldsymbol{x} - \boldsymbol{p}\|$, is denoted by a red-colored line. The Dijkstra geodesic distance is computed along the shortest path between x and p.

We incorporate the geodesic kernel matrix used in our earlier work [7, 8] into the current soft kernel k-means algorithm. The geodesic kernel matrix $\boldsymbol{K}$ is constructed in the following way:

- Construct a neighborhood graph where edge weights between only connected nodes (data points) are set as their Euclidean distances.
- Compute geodesic distances, D_{ij}, that are associated with the sum of edge weights along shortest paths between all pairs of points and define $\boldsymbol{D}^2 = \left[D_{ij}^2\right] \in \mathbb{R}^{N \times N}$.
- Construct a matrix $\widetilde{\boldsymbol{K}}(\boldsymbol{D}^2) = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{H}$, where $\boldsymbol{H}$ is the centering matrix given by $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{e}_N\boldsymbol{e}_N^\top$ for $\boldsymbol{e}_N = [1 \ldots 1]^\top \in \mathbb{R}^N$.
- Compute the largest eigenvalue, c^*, of the matrix

$$\begin{bmatrix} \boldsymbol{0} & 2\widetilde{\boldsymbol{K}}(\boldsymbol{D}^2) \\ -\boldsymbol{I} & -4\widetilde{\boldsymbol{K}}(\boldsymbol{D}) \end{bmatrix}. \tag{5}$$

- The geodesic kernel matrix is of the form

$$\boldsymbol{K} = \widetilde{\boldsymbol{K}}(\boldsymbol{D}^2) + 2c\widetilde{\boldsymbol{K}}(\boldsymbol{D}) + \frac{1}{2}c^2\boldsymbol{H}, \tag{6}$$

where $\boldsymbol{K}$ is guaranteed to be positive semidefinite (i.e., satisfies Mercer condition) for $c \geq c^*$.

Algorithm outline: Soft geodesic kernel k-means

Step 1. (Initialization) Given the number of clusters, k, initialize clusters $\{\mathcal{S}_j\}_{j=1}^{K}$ and responsibilities R_{jt} randomly, and construct the geodesic kernel matrix, $\boldsymbol{K}$ in (6).

Step 2. (Responsibilities) For each data point $\boldsymbol{x}_t$ $(t = 1, 2, \ldots, N)$, update responsibilities R_{jt} $(j = 1, \ldots, K)$ by

$$R_{jt} = \frac{\exp\left\{-\beta\left\|\phi(\boldsymbol{x}_t) - \boldsymbol{\mu}_j\right\|^2\right\}}{\sum_{i=1}^{K}\exp\left\{-\beta\left\|\phi(\boldsymbol{x}_t) - \boldsymbol{\mu}_i\right\|^2\right\}}, \tag{7}$$

where $\left\|\phi(\boldsymbol{x}_t) - \boldsymbol{\mu}_j\right\|^2$ is computed using the relation in (4).

Step 3. (Update) Determine cluster indices $\widehat{j}_t$ associated with $\boldsymbol{x}_t$ by solving

$$\widehat{j}_t = \arg\max_j R_{jt}, \tag{8}$$

and update clusters by

$$\mathcal{S}_j = \left\{\boldsymbol{x}_t \,|\, \widehat{j}_t = j\right\}, \quad j = 1, \ldots, K. \tag{9}$$

Step 4. (Repetition) Repeat Steps 2 and 3, until the convergence is achieved or the maximum number of iterations (pre-specified in advance) is reached.

3. NUMERICAL EXPERIMENTS

In this section, we show the usefulness of the soft geodesic kernel k-means algorithm, through the empirical comparison to the only soft kernel k-means with RBF kernel. We applied our algorithm to some artificial data sets, and carried out some experiments with real world examples.

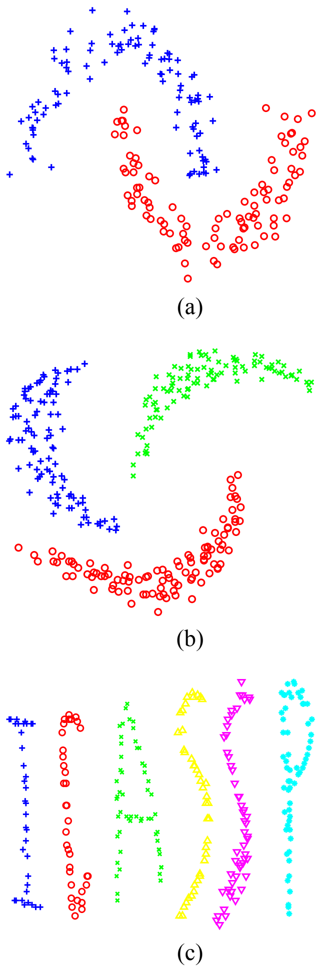
3.1. Experiment 1: Artificial data sets

We did numerical experiments with three different artificial data sets, and compared with the different kinds of kernel k-means. Figs. 2 shows clustering results. The first data set is 'Two moons' data where each moon consists of 104 and 96 data points, and the second data set is 'Three moons' data has three classes of 100 data points each. The last data set is 'ICASSP' log which has six groups where each letter indicates one cluster. Each cluster has 35, 35, 56, 41, 53, and 60 data points respectively. As we easily notice in Fig 2, these

Table 1. Results of clustering in terms of the classification accuracy (%, mean of each accuracy) according to a variety of parameter values (kernel size and neighborhood size) are summarized about two methods (i.e., ours and the soft kernel k-means with RBF kernel function) (we used $\beta = 0.6, 0.03$ and 0.03 in each softmax function).

method \ neighborhood size		4	5	6	7	8	9	10	11	12
ours	Two moons	**100**	**100**	**100**	**100**	**100**	**100**	**100**	94.792	92.188
	Three moons	**100**	**100**	**100**	**100**	99.667	99.333	99.333	99.333	99.333
	'ICASSP' logo	**100**	**100**	**100**	**100**	**100**	**100**	**100**	83.333	81.899
method \ kernel size		0.32	0.55	0.71	0.84	0.95	1.05	1.140	1.22	1.30
soft k-means with RBF kernel	Two moons	**100**	75.321	73.397	72.917	71.955	70.994	70.994	70.994	70.994
	Three moons	78.000	81.000	89.333	**99.667**	95.667	96.000	96.333	97.667	90.333
	'ICASSP' logo	50.804	82.858	**93.642**	78.625	58.908	67.072	71.715	71.715	59.048

problems cannot be clustered well using any density-based clustering algorithms. In addition, these data sets are composed of data points which are close to the other data points in Euclidean space, but these points are far away in geodesic space. Therefore, the use of geodesic kernel instead of the other general kernel functions is more reasonable in our approach.


(a)

(b)

(c)

Fig. 2. Clustering results for three data sets with our *soft geodesic kernel k-means*, is shown. For *two moons, three moons* data clustering result are shown in (a) and (b) respectively. For *'ICASSP'* logo data set, clustering result is shown in (c).

The soft kernel k-means with RBF kernel shows not bad clustering performances under certain parameter value as shown in Table 1. However the selection of kernel size is very critical, it is hard to find a kernel size for the desired clustering result.

Table 1 presents the clustering performances of two methods, according to various kernel sizes and neighborhood sizes respectively, which shows that our method has more stable results.

3.2. Experiment 2: Real-world data sets

In the first, as a real-world problem, we applied our soft geodesic kernel k-means to the task of document clustering that has played an important role in text information retrival. In here, we performed some experiments with '20 newsgroup'[1] data set which contains about 20,000 articles (see Table 2).

We selected top 1000 words by ranking the values of mutual information between terms and documents. The tf.idf (term frequency-inverse document frequency) scheme for term is used for constructing term-document matrix, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n \times d}$ where $\boldsymbol{x}_i$ indicates a document, n is the number of documents and d represents the number of being selected terms. The distance measure between two documents is defined as

$$Dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \frac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{\|\boldsymbol{x}_i\|\|\boldsymbol{x}_j\|}. \tag{10}$$

We focus on two data sets, each has three clusters. In the first experiment (three groups, NG1/NG2/NG15), we chose 100, 125, and 140 articles randomly from NG1, NG2, and NG15 respectively. In the second (three groups, NG2/NG10/NG18), we selected 100 articles randomly from each newsgroup. For each case, we preformed 10 independent experiments and its averaged results are summarized in Table 3. The clustering performance was measured in terms of classification accuracy, indicating how many documents were correctly classified.

In the second, we applied Iris and Wind data which are available from UCI Repository [10]. Iris data contain three groups of 50 instances each where each instance consists of 4 dimensional point. Wine data are 13 dimensional points in three classes whose sizes are 59, 71, and 48 respectively. In the case of Wine data, after whitening all points to zero mean and unity variance due to the fact of its heterogeneous attributes, we did our experiments. We preformed 10 independent experiments and its averaged results are summarized too.

As shown in Table 3, our method shows some improvement over the soft kernel k-means with RBF kernel. Moreover, all standard

[1] Dataset and the bow toolkit required to construct a term-document matrix, are available online [9].

Table 2. 20 newsgroup data and their indexing.

NG1	alt.atheism	NG11	rec.sport.hockey
NG2	comp.graphics	NG12	sci.crypy
NG3	comp.os.ms-windows.misc	NG13	sci.electronics
NG4	comp.sys.ibm.pc.hardware	NG14	sci.med
NG5	comp.sys.mac.hardware	NG15	sci.space
NG6	comp.windows.x	NG16	soc.religion.christian
NG7	misc.forsale	NG17	talk.politics.guns
NG8	rec.autos	NG18	talk.politics.mideast
NG9	rec.motorcycles	NG19	talk.politics.misc
NG10	rec.sport.baseball	NG20	talk.religion.misc

Table 3. Results of document and UCI data sets clustering in terms of the classification accuracy (%) are summarized for two experiments (we used $\beta = 0.8, 0.8, 0.6$ and 0.03 in each softmax function). Values in parenthesis represent standard deviation.

method \ newsgroup	NG1	NG2	NG15	Total (%)
ours (neighborhood size 45)	**90.625 (± 2.872)**	**88.750 (±2.363)**	**76.071 (±6.133)**	**85.148 (±2.479)**
soft k-means with RBF kernel (kernel size 0.894)	88.500 (±2.572)	88.266 (±6.579)	71.944 (±7.332)	82.903 (±4.635)
method \ newsgroup	NG2	NG10	NG18	Total (%)
ours (neighborhood size 80)	92.375 (±3.292)	**87.500 (±3.585)**	**91.375 (±2.199)**	**90.416 (±1.231)**
soft k-means with RBF kernel (kernel size 0.547)	**93.500 (±1.069)**	83.375 (±4.033)	87.875 (±8.166)	88.250 (± 3.650)
method \ Iris data set	class 1	class 2	class 3	Total (%)
ours (neighborhood size 26)	**100.000 (±0.000)**	82.000 (±0.000)	**98.000 (±0.000)**	**93.333 (±0.000)**
soft k-means with RBF kernel (kernel size 0.948)	**100.000 (±0.000)**	**85.466 (±11.915)**	85.200 (±12.393)	90.222 (±0.860)
method \ Wine data set	class 1	class 2	class 3	Total (%)
ours (neighborhood size 28)	**92.542 (± 3.591)**	**86.056 (±4.426)**	**96.250 (±1.914)**	**91.616 (±2.116)**
soft k-means with RBF kernel (kernel size 1.581)	81.355 (±5.282)	75.176 (±6.856)	83.072 (±10.706)	79.868 (±6.334)

deviation values of the total performances in our method are smaller than values of the counterpart method, which is due to the empirical results of its all responsibilities are near either 0 or 1. That is, we can notice that the performance of our method is more stable and prominent.

4. CONCLUSIONS

We have presented a clustering method, "soft geodesic kernel k-means", where we kernerlized the soft k-means algorithm, employing a geodesic kernel which reflected the data manifold. Useful aspects of our proposed clustering method could be summarized as follows: (a) It is simple but the kernel trick and soft decision allows us to identify non-convex clusters; (b) The geodesic kernel reflects a data manifold structure, which improves the clustering performance. Empirical comparison with soft kernel k-means with Gaussian kernel (RBF kernel), confirmed the high performance of our method, in the case of toy data sets as well as real-world problems.

Acknowledgments: This work was supported by Korea MIC under ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0045) and by National Core Research Center for Systems Bio-Dynamics.

5. REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiely & Sons, 2001.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[3] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1540–1551, 2003.

[4] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, 2002.

[5] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means, spectral clustering and normalized cuts," in *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, Seattle, WA, 2004.

[6] D. J. C. MacKay, *Information Theory, Inferecne, and Learning Algorithms*, Cambridge University Press, 2003.

[7] H. Choi and S. Choi, "Kernel Isomap on noisy manifold," in *Proc. Int'l Conf. Development and Learning*, Osaka, Japan, 2005.

[8] H. Choi and S. Choi, "Robust kernel Isomap," *Pattern Recognition*, vol. 40, no. 3, pp. 853–862, 2007.

[9] A. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," 1996.

[10] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998.