Journal of Advanced Computational Intelligence and Intelligent Informatics.
vol.15, no.7, pp.800-805, 2011.

Dependence-Maximization Clustering with Least-Squares Mutual Information

Manabu Kimura
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan.
kimura@sg.cs.titech.ac.jp

Masashi Sugiyama
Tokyo Institute of Technology,
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan.
sugi@cs.titech.ac.jp http://sugiyama-www.cs.titech.ac.jp/~sugi

Abstract

Recently, statistical dependence measures such as mutual information and kernelized covariance have been successfully applied to clustering, called dependence-maximization clustering. In this paper, we propose a novel dependence-maximization clustering method based on an estimator of a squared-loss variant of mutual information called least-squares mutual information. A notable advantage of the proposed method over existing ones is that hyperparameters such as kernel parameters and regularization parameters can be objectively optimized based on cross-validation. Thus, subjective manual-tuning of hyperparameters is not necessary in the proposed method, which is a highly useful property in unsupervised clustering scenarios. Through experiments, we illustrate the usefulness of the proposed approach.

Keywords

Dependence-maximization clustering, Squared-loss mutual information, Least-squares mutual information, Model selection, Structured data, Kernel

1 Introduction

Given a set of observations, the goal of clustering is to separate them into disjoint clusters so that observations in the same cluster are qualitatively similar to each other. *K-means* [22] is a classic clustering algorithm which minimizes the within-cluster distortion in a greedy manner. Although k-means is still a popular clustering method, it has a critical limitation that cluster boundaries are linear.

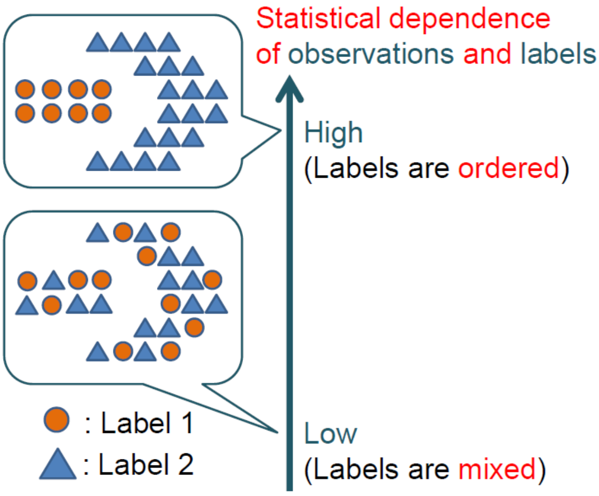

Figure 1: Schematic illustration of dependence-maximization clustering.

To overcome this limitation, various non-linear clustering algorithms have been developed. *Spectral clustering* [25, 23] first applies a spectral embedding method to data samples and then performs k-means in the embedding space. *Kernel k-means* [11] first transforms data samples by a kernel function and then performs k-means in the kernel-induced feature space. Note that spectral clustering was shown to be equivalent to a weighted variant of kernel k-means with some specific kernel [6]. *Discriminative clustering* trains a discriminative classifier such as the support vector machine in an unsupervised manner [28, 2, 12]. *Dependence-maximization clustering* determines cluster assignments so that their statistical dependence on input data is maximized [26, 8]. Figure 1 shows a schematic illustration of dependence-maximization clustering. Existing methods use mutual information [8] and kernelized covariance [26] as dependence measures.

In this paper, we propose a novel dependence-maximization clustering algorithm. Our method uses an estimator of a squared-loss variant of mutual information called *least-squares mutual information* (LSMI) [27] as a dependency measure. A notable advantage of the proposed method is that tuning parameters can be objectively optimized based on cross-validation. Thus, subjective manual-tuning of hyperparameters is not necessary in the proposed method, which is a highly useful property in unsupervised clustering scenarios. Through experiments, we illustrate the usefulness of the proposed approach.

The rest of this paper is structured as follows. In Section 2, we describe the proposed algorithm. In Section 3, we discuss the relation between the proposed and existing dependence-maximization clustering algorithms. In Section 4, experimental performance of the proposed and existing methods is compared. Finally, in Section 5, this paper is concluded.

2 Dependence-Maximization Clustering

In this section, we formulate the problem of dependence-maximization clustering and describe our proposed approach.

2.1 Problem Formulation

Given n i.i.d. observations $x_1, \ldots, x_n$, the goal of clustering is to assign a cluster label $y_i \in \{1, \ldots, c\}$ to each x_i, where c denotes the number of clusters. In this paper, we focus on the *dependence-maximization* framework of clustering, i.e., the ideal cluster assignments $y_1^*, \ldots, y_n^*$ are defined as the ones that have the maximum dependency to the observations $x_1, \ldots, x_n$.

As a dependency measure, we use a *squared-loss mutual information* (SMI) defined and expressed by

$$\mathrm{SMI} := \frac{1}{2} \int \sum_{y=1}^{c} \left(\frac{p(x,y)}{p(x)p(y)} - 1 \right)^2 p(x)p(y) dx \tag{1}$$

$$= \frac{1}{2} \int \sum_{y=1}^{c} p(x,y) \frac{p(x,y)}{p(x)p(y)} dx - \frac{1}{2}, \tag{2}$$

where $p(x,y)$, $p(x)$, and $p(y)$ denote the joint and marginal densities/probabilities of x and y. SMI is non-negative and takes zero if and only if x and y are statistically independent.

Note that SMI is the *Pearson divergence* [24] from $p(x,y)$ to $p(x)p(y)$, while the ordinary mutual information [4], defined by

$$\mathrm{MI} := \int \sum_{y=1}^{c} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dx, \tag{3}$$

is the *Kullback-Leibler divergence* [20] from $p(x,y)$ to $p(x)p(y)$. The Pearson divergence and the Kullback-Leibler divergence both belong to the class of *Ali-Silvey-Csiszár divergences* (also known as f-divergences, see [1, 5]), which share similar properties.

Since $p(x,y)$, $p(x)$, and $p(y)$ included in SMI are unknown, we cannot directly compute SMI. Our basic idea is to approximate SMI from the paired samples $(x_1, y_1), \ldots, (x_n, y_n)$, where $y_1, \ldots, y_n$ are hypothetical cluster assignments for the observations $x_1, \ldots, x_n$. Then the maximizers of the SMI approximator with respect to $y_1, \ldots, y_n$ are obtained as clustering results.

2.2 SMI Approximation by LSMI

The SMI approximator we use in this paper is called *least-squares mutual information* (LSMI) [27], which was shown to possess the optimal non-parametric convergence rate. Here we briefly review LSMI.

The key idea of LSMI is to learn the following *density-ratio function*,

$$r(x,y) := \frac{p(x,y)}{p(x)p(y)}, \tag{4}$$

without going through density/probability estimation of $p(x,y)$, $p(x)$, and $p(y)$. More specifically, we approximate the above density-ratio function by

$$\sum_{i=1}^{n} \theta_i K(x,x_i) L(y,y_i), \tag{5}$$

where $K(x,x')$ is a kernel function for x and $L(y,y')$ is a kernel function for y. The parameters $\theta_1,\ldots,\theta_n$ are learned so that the following squared error is minimized:

$$\frac{1}{2}\int \sum_{y=1}^{c} \left(\sum_{i=1}^{n} \theta_i K(x,x_i)L(y,y_i) - r(x,y) \right)^2 p(x)p(y)dx. \tag{6}$$

An empirical approximation of Eq.(6) is given as

$$\frac{1}{2n^2} \sum_{i,j,k,l=1}^{n} \theta_k \theta_l K(x_i,x_k)K(x_i,x_\ell)L(y_j,y_k)L(y_j,y_\ell) \tag{7}$$

$$- \frac{1}{n} \sum_{i,j=1}^{n} \theta_j K(x_i,x_j)L(y_i,y_j) + \text{Const.} \tag{8}$$

$$= \frac{1}{2}\boldsymbol{\theta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\theta} - \boldsymbol{\theta}^\top \widehat{\boldsymbol{h}} + \text{Const.}, \tag{9}$$

where $\top$ denotes the transpose, and $\widehat{\boldsymbol{H}}$ is the $n \times n$ matrix and $\widehat{\boldsymbol{h}}$ is the n-dimensional vector defined as

$$\widehat{H}_{\ell,\ell'} := \frac{1}{n^2} \sum_{i,j=1}^{n} K(x_i,x_\ell)K(x_i,x_{\ell'})L(y_j,y_\ell)L(y_j,y_{\ell'}), \tag{10}$$

$$\widehat{h}_\ell := \frac{1}{n} \sum_{i=1}^{n} K(x_i,x_\ell)L(y_i,y_\ell). \tag{11}$$

Further adding a regularization term, we arrive at the following optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\boldsymbol{\theta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\theta} - \boldsymbol{\theta}^\top \widehat{\boldsymbol{h}} + \lambda \boldsymbol{\theta}^\top \boldsymbol{\theta}, \tag{12}$$

where λ (≥ 0) is the regularization parameter. The solution $\widehat{\boldsymbol{\theta}}$ can be computed analytically as

$$\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I})^{-1} \widehat{\boldsymbol{h}}, \tag{13}$$

where $\boldsymbol{I}$ denotes the identity matrix. Then a density-ratio estimator is obtained as

$$\widehat{r}(x,y)=\sum_{i=1}^{n}\widehat{\theta}_i K(x,x_i)L(y,y_i). \tag{14}$$

Finally, an SMI estimator called LSMI is given as

$$\mathrm{LSMI}:=\frac{1}{2n}\sum_{i,j=1}^{n}\widehat{\theta}_i K(x_i,x_j)L(y_i,y_j)-\frac{1}{2}. \tag{15}$$

In experiments, we use the *delta kernel* as $L(y,y')$, i.e.,

$$L(y,y')=\begin{cases}1 & (y=y'),\\ 0 & (y\neq y').\end{cases} \tag{16}$$

Then, the matrix $\widehat{\boldsymbol{H}}$ becomes block-diagonal, given that the observations $x_1,\ldots,x_n$ are sorted according to the cluster assignments. Thus, the matrix inversion in Eq.(13) can be computed efficiently.

A MATLAB implementation of LSMI is available from '`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSMI/index.html`'.

2.3 Hyperparameter Choice by CV

The accuracy of the above least-squares density-ratio estimator depends on the choice of the hyperparameters such as the regularization parameter λ and some parameters included in the kernel functions $K(x,x')$ and $L(y,y')$. They can be systematically optimized based on cross-validation (CV) as follows [27].

The samples $\mathcal{Z}=\{(x_1,y_1),\ldots,(x_n,y_n)\}$ are divided into M disjoint subsets $\mathcal{Z}_1,\ldots,\mathcal{Z}_M$ of approximately the same size. Then a density-ratio estimator $\widehat{r}_m(x,y)$ is obtained using $\mathcal{Z}\backslash\mathcal{Z}_m$ (i.e., all samples without $\mathcal{Z}_m$), and its out-of-sample error for the hold-out samples $\mathcal{Z}_m$ is computed as

$$\frac{1}{2|\mathcal{Z}_m|^2}\sum_{x,y\in\mathcal{Z}_m}\widehat{r}_m(x,y)^2-\frac{1}{|\mathcal{Z}_m|}\sum_{(x,y)\in\mathcal{Z}_m}\widehat{r}_m(x,y). \tag{17}$$

This procedure is repeated for $m=1,\ldots,M$, and the average of the above hold-out error over all m is computed. Finally, the hyperparameters that minimize the average hold-out error are chosen.

2.4 Proposed Algorithm: LSMI Clustering

We determine the cluster assignments $y_1,\ldots,y_n$ so that the above LSMI is maximized. This is carried out in a greedy manner as follows.

1. Initialize $y_1, \ldots, y_n$ for $x_1, \ldots, x_n$.
2. For $i = 1, \ldots, n$, update y_i so that LSMI is maximized.
3. Repeat 2 until $y_1, \ldots, y_n$ do not change.

We call the above clustering algorithm *LSMI clustering* (LSMIC).

3 Related Work

In this section, we discuss the relation between the proposed and existing dependence-maximization clustering algorithms.

3.1 CLUHSIC

The *Hilbert-Schmidt independence criterion* (HSIC) [13] is a kernel-based dependence measure. Based on HSIC, a dependence-maximization clustering method called *clustering with HSIC* (CLUHSIC) was proposed [26].

CLUHSIC tries to determine cluster assignments $y_1, \ldots, y_n$ in a greedy manner so that HSIC is maximized[1]:

$$\mathrm{HSIC} = \sum_{i,j=1}^{n} \bar{K}(x_i, x_j) L(y_i, y_j), \tag{18}$$

where $\bar{K}(x, x')$ is a *centered* kernel.

If we ignore irrelevant constants in LSMI, it is expressed as

$$\sum_{i,j=1}^{n} \hat{\theta}_i K(x_i, x_j) L(y_i, y_j). \tag{19}$$

This shows that HSIC and LSMI are quite similar to each other. Their differences in appearance are

- The kernel $K(x, x')$ is not centered in LSMI.
- The summation of kernels is weighted according to $\hat{\theta}_1, \ldots, \hat{\theta}_n$ in LSMI.

However, a more essential difference lies in hyperparameter choice. LSMI is equipped with CV. Therefore, all the tuning parameters can be objectively optimized. On the other hand, there is no systematic model selection procedure for HSIC. Using the Gaussian kernel with width set to the input dimensionality or the median distance between samples is a standard heuristic in practice [13, 26]. As we will experimentally show in Section 4, this heuristic works reasonably well. However, this heuristic is not applicable to other

[1] `http://www.cs.cmu.edu/~lesong/code/cluhsic.zip`

kernels such as string kernels, tree kernels, and graph kernels [21, 7, 15, 18, 16, 10, 9]. Thus, when structured data are clustered, kernel parameters need to be tuned manually, which is highly subjective in unsupervised clustering scenarios. See Section 4 for more details.

3.2 NIC

Another dependence-maximization clustering method called *nonparametric information clustering* (NIC) adopts mutual information as a dependency measure [8].

NIC is based on the k-nearest neighbor entropy estimator [19]. The performance of the original k-nearest neighbor entropy estimator depends on the choice of the number of nearest neighbors, k. On the other hand, NIC avoids this problem by introducing a heuristic of taking an average over all possible k. The resulting objective function is given by

$$\sum_{y=1}^{c} \frac{1}{n_y - 1} \sum_{i \neq j : y_i = y_j = y} \log(\|x_i - x_j\| + \varepsilon), \tag{20}$$

where n_y denotes the number of samples in cluster y, and ε (> 0) is a smoothing parameter. This objective function is minimized with respect to cluster assignments $y_1, \ldots, y_n$ using a greedy algorithm.

Although the fact that the tuning parameter k is averaged out is practically convenient, this heuristic is not well justified. Moreover, the choice of the smoothing parameter ε is arbitrary. In the program code[2] provided by one of the authors, $\varepsilon = 1/n$ was recommended. However, there seems no justification for this choice.

Let $\delta(y, y')$ be the *Dirac delta*, i.e,

$$\delta(y, y') = \begin{cases} 1 & (y = y'), \\ 0 & (y \neq y'). \end{cases} \tag{21}$$

Then the above NIC criterion can be expressed as

$$\sum_{i \neq j} \frac{1}{n_{y_i} - 1} \log(\|x_i - x_j\| + \varepsilon) \delta(y_i, y_j). \tag{22}$$

Thus, if we relate $\frac{1}{n_{y_i}-1}$ to $\hat{\theta}_i$, $\log(\|x_i - x_j\| + \varepsilon)$ to $K(x_i, x_j)$, and $\delta(y_i, y_j)$ to $L(y_i, y_j)$, the appearance of the NIC criterion is rather similar to LSMI. However, there are critical differences between LSMI and NIC. The most critical one is that there is no systematic method to choose the hyperparameter ε in the NIC criterion, while LSMI is equipped with CV. Another difference is that any kernels can be used as $K(x, x')$ and $L(y, y')$ in LSMI, while they are restricted to specific ones in the NIC criterion.

[2] `http://www.levfaivishevsky.webs.com/NIC.rar`

By using

$$\|x_i - x_j\| = \sqrt{\|x_i\|^2 + \|x_j\|^2 - 2x_i^\top x_j}, \tag{23}$$

the kernel trick can be employed. Thus, in principle, NIC is applicable to structured data. However, this uses kernels on 'kernel' $\log(\|x_i - x_j\| + \varepsilon)$, and its validity is unclear. Furthermore, lack of hyperparameter selection methods is again a critical limitation when structured data are clustered. See Section 4 for more details.

4 Experiments

In this section, we experimentally compare the clustering performance of LSMIC with that of CLUHSIC and NIC.

First, we employ some of the *UCI benchmark datasets*[3]. These are classification datasets with vectorial features. The specification of the datasets is described in the left column of Table 1. The Gaussian kernel is used for LSMIC, where the kernel width is chosen by CV. We also use the Gaussian kernel for CLUHSIC, but it requires the user to specify the kernel width manually. There seem two popular heuristics for the kernel width choice—using the feature dimensionality [26] or the median distance between samples [13] as the Gaussian width. Here we test both heuristics, which are indicated by 'CLUHSIC(dim)' and 'CLUHSIC(med)', respectively. The smoothing parameter ε in NIC is fixed to $1/n$, following the suggestion by the authors. Each method is executed 9 times, and the best result in terms of each objective value is chosen. Before feeding the data into each algorithm, we normalize the data so that element-wise variance is one. For NIC, we further whiten the data, as suggested in [8]. The experimental results are summarized in Table 1, showing that all the methods work comparably well for these simple tasks.

Next, we consider clustering tasks for structured data. We use the *Brown corpus dataset*[4], which is a carefully compiled selection of current American English. It consists of a million words sampled from 15 genres such as news and religion, and is accompanied with part-of-speech tags which represent relationship with adjacent and related words in a phrase, sentence, or paragraph. We convert the Brown corpus data to *dependence tree* representation by the *MaltParser*[5].

As kernel functions, we use a version of the *labeled ordered tree kernel* [15] between two dependence trees, which counts the number of sub-trees common to both trees. The similarity between nodes is computed as the inner product of vectors $(p(z_1|w), \ldots, p(z_{20}|w))$, where $\{p(z_i|w)\}_{i=1}^{20}$ are the probabilities of topic z_i under word w calculated by *probabilis-*

[3] The UCI benchmark datasets are available from `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[4] The Brown corpus dataset can be downloaded using the *Natural Language Toolkit* (`http://www.nltk.org/`), which contains open source Python modules, linguistic data, and documentation for research and development in natural language processing and text analysis.

[5] The MaltParser is available from `http://maltparser.org/`.

Table 1: Experimental results on UCI datasets. The average clustering accuracy (and its standard deviation in the bracket) over 100 runs is described. Higher accuracy is better. The best method in terms of the average accuracy and methods judged to be comparable to the best one by the t-test at the significance level 1 % are described in boldface.

Dataset $< \#\text{Dim.}, \#\text{Class} >$	LSMIC	CLUHSIC(dim)	CLUHSIC(med)	NIC
Acoustic $< 50, 3 >$	**54.5** (6×10^{-2})	**55.0** (6×10^{-2})	**54.9** (6×10^{-2})	**53.0** (6×10^{-2})
Seismic $< 50, 3 >$	58.4 (5×10^{-2})	**60.3** (4×10^{-2})	**59.7** (4×10^{-2})	57.7 (5×10^{-2})
Sonar $< 60, 2 >$	**57.4** (5×10^{-2})	**56.6** (4×10^{-2})	**56.9** (4×10^{-2})	**56.8** (4×10^{-2})
Transfusion $< 4, 2 >$	**58.7** (5×10^{-2})	**58.1** (4×10^{-2})	**58.3** (5×10^{-2})	**58.7** (6×10^{-2})
Madelon $< 500, 2 >$	**56.6** (4×10^{-2})	**56.6** (5×10^{-2})	**56.9** (4×10^{-2})	**56.5** (4×10^{-2})
Iris $< 4, 3 >$	81.1 (4×10^{-2})	**83.9** (3×10^{-2})	**83.9** (2×10^{-2})	**83.0** (4×10^{-2})
Haberman $< 3, 2 >$	**54.8** (4×10^{-2})	**54.2** (3×10^{-2})	**54.2** (3×10^{-2})	**54.3** (3×10^{-2})
Pima $< 8, 2 >$	**65.3** (7×10^{-2})	**66.9** (5×10^{-2})	**68.1** (6×10^{-2})	**67.0** (6×10^{-2})

Table 2: Experimental results on the Brown corpus dataset using labeled ordered tree kernels. The average clustering accuracy (and its standard deviation in the bracket) over 100 runs is described. Higher accuracy is better. The best method in terms of the average accuracy and methods judged to be comparable to the best one by the t-test at the significance level 1 % are described in boldface.

Adventure vs.	LSMIC	CLUHSIC			NIC		
other topics		$\kappa = 0.1$	$\kappa = 0.4$	$\kappa = 0.7$	$\kappa = 0.1$	$\kappa = 0.4$	$\kappa = 0.7$
Belles_letteres	**73.6**	**71.9**	**71.3**	56.8	**71.8**	69.2	58.0
	(8×10^{-2})	(7×10^{-2})	(6×10^{-2})	(5×10^{-2})	(6×10^{-2})	(7×10^{-2})	(6×10^{-2})
Editional	**65.0**	58.6	58.4	57.8	**61.0**	**61.5**	57.2
	(1×10^{-1})	(8×10^{-2})	(8×10^{-2})	(5×10^{-2})	(1×10^{-1})	(1×10^{-1})	(5×10^{-2})
Fiction	**54.3**	**54.2**	**54.4**	**54.3**	**54.4**	**54.2**	**54.5**
	(3×10^{-2})	(3×10^{-2})	(3×10^{-2})	(3×10^{-2})	(3×10^{-2})	(3×10^{-2})	(3×10^{-2})
Government	**81.8**	69.4	71.5	66.5	70.7	69.7	62.1
	(2×10^{-1})	(2×10^{-1})	(2×10^{-1})	(1×10^{-1})	(2×10^{-1})	(2×10^{-1})	(1×10^{-1})
Hobbies	**70.9**	64.0	64.9	60.1	**67.8**	65.9	60.3
	(1×10^{-1})	(1×10^{-1})	(1×10^{-1})	(7×10^{-2})	(1×10^{-1})	(1×10^{-1})	(8×10^{-2})
Humor	60.0	**63.3**	**61.9**	58.8	60.7	59.7	57.7
	(4×10^{-2})	(5×10^{-2})	(5×10^{-2})	(5×10^{-2})	(5×10^{-2})	(5×10^{-2})	(7×10^{-2})
Learned	**89.5**	86.4	86.7	59.1	**87.7**	83.5	60.7
	(5×10^{-2})	(9×10^{-2})	(8×10^{-2})	(8×10^{-2})	(6×10^{-2})	(8×10^{-2})	(9×10^{-2})

tic latent semantic indexing [14][6]. The inner product has a high value if the words share a similar topic. We then normalize this similarity value and the kernel value [17].

The labeled ordered tree kernel contains a tuning parameter, the *decay factor* κ $(0 < \kappa \leq 1)$ which controls the weights for large sub-trees [3]. We choose κ from $\{0.1, 0.4, 0.7\}$ by CV for LSMIC. On the other hand, there is no systematic way to choose κ for CLUHSIC and NIC, so we test all three cases. We perform clustering between the topic 'Adventure' and one of the other topics: 'Belles_letteres', 'Editional', 'Fiction', 'Government', 'Hobbies', 'Humor', and 'Learned'. The results are described in Table 2, showing that LSMIC overall compares favorably with CLUHSIC and NIC.

5 Conclusion

In this paper, we proposed a novel dependence-maximization clustering method. Our method used *least-squares mutual information*, an optimal non-parametric estimator of a squared-loss variant of mutual information, as a dependency measure. A notable advantage of LSMI is that hyperparameters such as kernel parameters and regularization parameters can be objectively optimized based on cross-validation. Thanks to this, subjective manual-tuning of hyperparameters is not necessary in the proposed method. In practice, this is a highly useful property in unsupervised clustering scenarios. Through experiments, we illustrated the usefulness of the proposed approach.

Similarly to other clustering approaches, initialization of cluster assignments is a key issue in the proposed LSMI clustering algorithm. This needs to be addressed in the future work.

Acknowledgements

This work was partially supported by SCAT, AOARD, and the JST PRESTO program.

References

[1] S. M. Ali and S. D. Silvey, "A General Class of Coefficients of Divergence of One Distribution from Another", Journal of the Royal Statistical Society, Series B, 28(1):131–142, 1966.

[2] F. Bach and Z. Harchaoui, "DIFFRAC: a discriminative and flexible framework for clustering", In Advances in Neural Information Processing Systems 20 (NIPS 2007), pp. 49–56, 2008.

[3] M. Collins and N. Duffy, "Convolution Kernels for Natural Language", In Advances in Neural Information Processing Systems 14 (NIPS 2001), pp. 625–632, 2002.

[6] A software of probabilistic latent semantic indexing is available from `http://chasen.org/~taku/software/plsi/plsi-0.03.tar.gz`

[4] T. M. Cover and J. A. Thomas, ”Elements of Information Theory”, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2nd edition, 2006.

[5] I. Csiszár, “Information-type Measures of Difference of Probability Distributions and Indirect Observation”, Studia Scientiarum Mathematicarum Hungarica, 2:229–318, 1967.

[6] I. S. Dhillon, Y. Guan, and B. Kulis, “Kernel K-means, Spectral Clustering and Normalized Cuts”, In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 551–556, 2004.

[7] N. Duffy and M. Collins, “Convolution Kernels for Natural Language”, In Advances in Neural Information Processing Systems 14 (NIPS 2001), pp. 625–632, 2002.

[8] L. Faivishevsky and J. Goldberger, “A Nonparametric Information Theoretic Clustering Algorithm”, In Proceedings of 27th International Conference on Machine Learning (ICML 2010), pp. 351–358, 2010.

[9] T. Gärtner, “A Survey of Kernels for Structured Data”, SIGKDD Explorations, 5(1):S268–S275, 2003.

[10] T. Gärtner, P. Flach, and S. Wrobel, “On Graph Kernels: Hardness Results and Efficient Alternatives”, In Proceedings of the 16th Annual Conference on Computational Learning Theory (COLT 2003), pp. 129–143, 2003.

[11] M. Girolami, “Mercer Kernel-Based Clustering in Feature Space”, IEEE Transactions on Neural Networks, 13(3):780–784, 2002.

[12] R. Gomes, A. Krause, and P. Perona, “Discriminative Clustering by Regularized Information Maximization”, In Advances in Neural Information Processing Systems 23 (NIPS 2010), pp. 766–774, 2010.

[13] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, “Measuring Statistical Dependence with Hilbert-Schmidt Norms”, In Proceedings of the 16th International Conference on Algorithmic Learning Theory (ALT 2005), Lecture Notes in Artificial Intelligence, pp. 63–77, 2005.

[14] T. Hofmann, “Probabilistic Latent Semantic Indexing”, In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999), pp. 50–57, 1999.

[15] H. Kashima and T. Koyanagi, “Kernels for Semi-Structured Data”, In Proceedings of the 19th International Conference on Machine Learning (ICML 2002), pp. 291–298, 2002.

[16] H. Kashima, K. Tsuda, and A. Inokuchi, “Marginalized Kernels between Labeled Graphs”, In Proceedings of the 20th International Conference on Machine Learning (ICML 2003), pp. 321–328, 2003.

[17] J. Kazama and K. Torisawa, “Speeding up Training with Tree Kernels for Node Relation Labeling”, In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), pp. 137–144, 2005.

[18] R. I. Kondor and J. Lafferty, “Diffusion Kernels on Graphs and Other Discrete Input Spaces”, In Proceedings of the 19th International Conference on Machine Learning (ICML 2002), pp. 315–322, 2002.

[19] L. F. Kozachenko and N. N. Leonenko, “Sample Estimate of Entropy of a Random Vector”, Problems of Information Transmission, 23(9):95–101, 1987.

[20] S. Kullback and R. A. Leibler, “On Information and Sufficiency”, Annals of Mathematical Statistics, 22:79–86, 1951.

[21] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text Classification Using String Kernels”, Journal of Machine Learning Research, 2:419–444, 2002.

[22] J. B. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations”, In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pp. 281–297, 1967.

[23] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On Spectral Clustering: Analysis and An Algorithm”, In Advances in Neural Information Processing Systems 14 (NIPS 2001), pp. 849–856, 2002.

[24] K. Pearson, “On the Criterion That a Given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such That It Can Be Reasonably Supposed to Have Arisen from Random Sampling”, Philosophical Magazine, 50:157–175, 1900.

[25] J. Shi and J. Malik, “Normalized Cuts and Image Segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

[26] L. Song, A. Smola, A. Gretton, and K. Borgwardt, “A Dependence Maximization View of Clustering”, In Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007), pp. 815–822, 2007.

[27] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, “Mutual Information Estimation Reveals Global Associations between Stimuli and Biological Processes”, BMC Bioinformatics, 10(1):S52, 2009.

[28] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, “Maximum Margin Clustering”, In Advances in Neural Information Processing Systems 17 (NIPS 2004), pp. 1537–1544, 2005.