Denoising Cluster Analysis

Ruqi Zhang, Zhirong Yang(✉), and Jukka Corander

Helsinki Institute for Information Technology HIIT,
University of Helsinki, Helsinki, Finland
zhirong.yang@helsinki.fi

Abstract. Clustering or cluster analysis is an important and common task in data mining and analysis, with applications in many fields. However, most existing clustering methods are sensitive in the presence of limited amounts of data per cluster in real-world applications. Here we propose a new method called denoising cluster analysis to improve the accuracy. We first construct base clusterings with artificially corrupted data samples and later learn their ensemble based on mutual information. We develop multiplicative updates for learning the aggregated cluster assignment probabilities. Experiments on real-world data sets show that our method unequivocally improves cluster purity over several other clustering approaches.

1 Introduction

Cluster analysis or clustering is an exploratory data analysis tool which aims at dividing data objects into groups such that objects in the same group are more similar than those in other groups. As one of the major tools for modern data mining and analysis, clustering research has found a wide variety of applications in many domains of science and technology.

Most clustering methods are built upon statistical laws, assuming a wealth of samples are available per cluster. With a limited amount of data points, many existing cluster analysis approaches can only achieve mediocre performance. Especially, methods that employ non-convex objectives are prone to yield poor local optima, which demands more complicated pre-training or initialization.

In this paper we propose a new clustering technique called denoising cluster analysis (DECLU). We first manually incorporate a small amount of noise among the data points. This is equivalent to sampling from the underlying smoothed data distribution, which potentially can generate infinite amounts of training data. We first build a base clustering for each noisy version of the data set. Next we aggregate the basal partitions into a single final clustering by using an information theoretic measure based on mutual information.

As an algorithmic contribution, we develop a new clustering ensemble method based on nonnegative learning, without construction of dense and expensive consensus relationship graph. We derive the multiplicative update rule for right stochastic matrices, which result in probabilistic cluster assignments.

We test the new method on various real-world data sets and compare it with several other clustering and ensemble clustering methods. The experimental

S. Arik et al. (Eds.): ICONIP 2015, Part III, LNCS 9491, pp. 435–442, 2015.
DOI: 10.1007/978-3-319-26555-1_49

results indicate that our method is more advantageous in terms of clustering accuracy.

The remaining paper is organized as follows. Section 2 briefly reviews mutual information and its application in comparing clusterings. Next we present our method in Sect. 3, with the base clustering generation, the ensemble clustering objective, and its optimization using multiplicative updates. In Sect. 4, we report the experiment setting and results. Section 5 summarizes the paper and discusses some possibilities for future work.

In what follows, a clustering is represented by a cluster indicator matrix, for example, denoted by U where $U_{ik} = 1$ if the ith sample is in the kth cluster, and $U_{ik} = 0$ elsewhere.

2 Preliminary: Mutual Information

In probability theory and information theory, the mutual information (MI) of two random variables is a measure of the mutual dependence between variables. The mutual information of two discrete random variables X and Y can be defined as

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}. \tag{1}$$

Mutual information measures the information that X and Y share. If X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero. At the other extreme, if X is a deterministic function of Y and Y is a deterministic function of X then all information conveyed by X is shared with Y. In this case the mutual information is the same as the uncertainty (entropy) contained in Y (or X) alone.

Mutual information (MI) can be used to compare two clusterings U and V (see e.g., [12]). Let n_{ij} be the number of objects that are common to the ith cluster in U and the jth cluster in V, $a_i = \sum_j n_{ij}$, $b_j = \sum_i n_{ij}$, and $\sum_{ij} n_{ij} = N$. Then

$$I(U;V) = \sum_i \sum_j \frac{n_{ij}}{N} \log \frac{n_{ij}/N}{a_i b_j / N^2}. \tag{2}$$

A larger MI indicates that the two clusterings are closer up to a certain cluster permutation. Various normalizations can be applied to fix the MI range in $[0,1]$. See [12] for a summary.

3 Denoising Clustering

Learning with artificially corrupted data, represented by training samples with manually incorporated noise, is a well-known trick in many machine learning settings, for example, generating additional training examples for Support Vector

Machine classifier to improve generalization performance (see e.g., [2,4]), and reconstructing input data from artificial corruption in Denoising Auto-Encoder for learning useful representations of data (see e.g., [10,11]). In this work, we apply a similar trick to cluster analysis. This encompasses two steps: we first construct base clusterings for noisy versions of the data, and then aggregate them into a single final clustering.

3.1 Generating Base Clusterings

A good clustering should respect the data distribution that underlies finite amount of sample data points. Kernel smoothing or Parzen window method [6] is a common approach to estimate the distribution that underlies the given data points.

In this work, instead of explicitly run kernel density estimation, which is usually expensive, we employ an implicit way to achieve a similar regularization for the clustering task for better accuracy. In the following, we implicitly use the Parzen window with Gaussian radial kernel, but the same technique can be extended to other kernels in a straightforward manner.

We add white noise to each data point x_i:

$$\tilde{x}_i = x_i + \epsilon_i, \tag{3}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma)$. Next we apply a relatively simple clustering method to partition the noisy data $\widetilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_N\}$. We choose Normalized Cut [8] for the base clusterings because it is less sensitive to the initializations and performs better for data in curved manifolds.

3.2 Clustering Ensemble Using Mutual Information

Next we find the consensus clustering of the basal partitions based on noisy versions of the original data. Classical clustering ensemble methods require construction of a pairwise relationship graph, which is quadratic to the number of samples and thus prohibitive for large-scale data sets. Here we propose a new method that directly learns the cluster assignment probabilities of size $N \times r$ for N data points and r clusters. This significantly reduces the computational cost.

Let $U^{(m)}$ denote the mth base clustering indicator matrix, where $m = 1, \ldots, M$. We seek a probabilistic cluster ensemble W where W_{ik} is the probability of the kth cluster given the ith sample. The ensemble minimizes the total difference to the base clusterings, measured by the "D_{sum}" distance based on mutual information [12]:

$$\underset{W \geq 0}{\text{minimize}} \quad \mathcal{J}(W) = \sum_m H(W) + H(U^{(m)}) - 2I(W; U^{(m)}) \tag{4}$$

$$\text{subject to} \quad \sum_{k=1}^{r} W_{ik} = 1, \; i = 1, \ldots, N, \tag{5}$$

where H is the entropy of cluster assignment. We choose this objective because it is a valid metric, bounded in $[0, M \log N]$, and with relatively simple gradient for optimization.

Writing out the objective, we have

$$\begin{aligned}
\mathcal{J}(W) = \quad & \sum_m \Bigg[-\sum_k \frac{1}{N} \sum_i W_{ik} \log \frac{1}{N} \sum_i W_{ik} \\
& -\sum_k \frac{1}{N} \sum_i U_{ik}^{(m)} \log \frac{1}{N} \sum_i U_{ik}^{(m)} \\
& -2 \sum_{kl} \frac{1}{N} \sum_i W_{ik} U_{il}^{(m)} \log \frac{\frac{1}{N} \sum_i W_{ik} U_{il}^{(m)}}{\frac{1}{N} \sum_i W_{ik} \frac{1}{N} \sum_i U_{il}^{(m)}} \Bigg] \\
= \quad & \sum_m \Bigg[\frac{1}{N} \sum_k \sum_i W_{ik} \log \sum_i W_{ik} \\
& - \frac{2}{N} \sum_{kl} \sum_i W_{ik} U_{il}^{(m)} \log \sum_i W_{ik} U_{il}^{(m)} \\
& + \frac{2}{N} \sum_i \left(\sum_k W_{ik} \right) \sum_l U_{il}^{(m)} \log \sum_j U_{jl}^{(m)} \Bigg] + \text{constant}. \qquad (6)
\end{aligned}$$

Using Lagrangian multipliers $\lambda = [\lambda_1, \ldots, \lambda_N]$ for the sum-to-one constraints, the relaxed objective function is

$$\widetilde{\mathcal{J}}(W, \lambda) = \mathcal{J}(W) - \sum_i \lambda_i (\sum_k W_{ik} - 1). \qquad (7)$$

Its gradient w.r.t. W is $\dfrac{\partial \widetilde{\mathcal{J}}(W)}{\partial W_{ik}} = \nabla_{ik}^{+} - \nabla_{ik}^{-} - \lambda_i$, where ∇^{+} and ∇^{-} are the positive and (unsigned) negative parts of the $\frac{\partial \mathcal{J}(W)}{\partial W}$

$$\nabla_{ik}^{+} = -\frac{2}{N} \sum_m \sum_t \left(\log \frac{\sum_i W_{ik} U_{it}^{(m)}}{N} \right) U_{it}^{(m)} + \frac{M}{N} (1 + \log N) \qquad (8)$$

$$\nabla_{ik}^{-} = -\frac{M}{N} \log \frac{\sum_i W_{ik}}{N} - \frac{2}{N} \sum_m \sum_l U_{il}^{(m)} \log \frac{\sum_j U_{jl}^{(m)}}{N}. \qquad (9)$$

This suggests the preliminary multiplicative update rule $W'_{ik} = W_{ik} \dfrac{\nabla_{ik}^{-} + \lambda_i}{\nabla_{ik}^{+}}$. Imposing the constraints $\sum_k W'_{ik} = 1$, we have $\displaystyle\sum_k W_{ik} \frac{\nabla_{ik}^{-}}{\nabla_{ik}^{+}} + \lambda_i \sum_k \frac{W_{ik}}{\nabla_{ik}^{+}} = 1$. Solving the equation we obtain

$$\lambda_i = \frac{1 - \sum_k W_{ik} \nabla_{ik}^{-} / \nabla_{ik}^{+}}{\sum_k W_{ik} / \nabla_{ik}^{+}} \qquad (10)$$

Putting them back to the preliminary rule, we have $W_{ik}^{'} = W_{ik} \frac{\nabla_{ik}^{-} A_{ik} + 1 - B_{ik}}{\nabla_{ik}^{+} A_{ik}}$, where

$$A_{ik} = \sum_k \frac{W_{ik}}{\nabla_{ik}^{+}} \text{ and } B_{ik} = \sum_k W_{ik} \frac{\nabla_{ik}^{-}}{\nabla_{ik}^{+}}. \tag{11}$$

There is a negative term $-B_{ik}$ in the numerator, which may cause negative entries in the updated W. To overcome this, we apply the "moving term" trick [15–18, 21] to resettle B_{ik} to the denominator, giving the final update rule

$$W_{ik}^{\text{new}} = W_{ik} \frac{\nabla_{ik}^{-} A_{ik} + 1}{\nabla_{ik}^{+} A_{ik} + B_{ik}}. \tag{12}$$

Our ensemble algorithm simply iterates the above update rule until W converges. The updates have the following guarantee

Theorem 1. *$\widetilde{\mathcal{J}}(W^{new}, \lambda) \leq \widetilde{\mathcal{J}}(W, \lambda)$, with λ given in Eq. 10.*

The proof is given in the Appendix. The theorem shows that the algorithm jointly reduces $\mathcal{J}(W)$ while steering W rows to the probability simplex. The tradeoff between these two forces is adaptively adjusted by A_{ik}.

4 Experiments

We have tested our new method on six real-world data sets from the UCI repository[1]. Their statistical characteristics are given in Table 1 and below a brief verbal description of each data set is given:

- ECOLI, the UCI *Ecoli* data set, containing protein localization sites, originally with 8 attributes.
- WINE, the UCI *Wine* data set, which is a result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars, originally with 13 dimensional features;
- MFEAT, the UCI *Multiple Features* data set, which consists of features of handwritten numerals; the digits are represented in terms of 649 features from six aspects;
- SEGMENT, the UCI *Image Segmentation* data set, image patches from 7 outdoor images, originally with 19 high-level features;
- OPTDIGITS, the UCI *optical recognition of handwritten digits* data set, originally with 64 dimensions;
- PENDIGITS, the UCI *pen-based recognition of handwritten digits* data set, originally with 16 dimensions.

[1] http://archive.ics.uci.edu/ml/.

Table 1. Statistics of the data sets.

Data set	Samples	Dimensions	Classes
ECOLI	327	7	5
WINE	178	13	3
MFEAT	2000	649	10
SEGMENT	2310	19	7
OPTDIGITS	5620	64	10
PENDIGITS	10992	16	10

We have compared DECLU with five other clustering approaches, three single clustering methods and two clustering ensemble methods:

- *Normalized Cut* (Ncut) [8], a spectral clustering method that projects the tailing eigenvectors of symmetric normalized Laplacian of the similarity matrix to the closest cluster indicator matrix;
- *Probabilistic Latent Semantic Indexing* (PLSI) [5] which factorize the similarity matrix $P(x_i, x_j) \approx \sum_k P(x_i|C = k)P(x_j|C = k)P(C = k)$, where C is the cluster variable; the cluster assignment $P(C = k|x_i)$ can then be obtained with $P(x_i|C = k)$ and $P(C = k)$ through Bayes rule;
- *Left Stochastic Decomposition* (LSD) [1] which factorizes the similarity matrix into two left-stochastic matrices;
- *Cluster-based similarity partitioning algorithm* (CSPA) [9], the similarity between two data-points is defined to be directly proportional to number of constituent clusterings of the ensemble in which they are clustered together;
- *Meta-clustering algorithm* (MCLA) [9], which is based on clustering clusters; first, it tries to solve the cluster correspondence problem and then uses voting to place data-points into the final consensus clusters.

We use the default setting in the above methods. The number of clusters is set to the number of known classes in each data set. We followed the convention that uses kmeans for generating base clustering for CSPA and MCLA. For DECLU,

Table 2. Clustering purities for the compared methods. Boldface numbers indicate the best for each data set.

Data set	Ncut	PLSI	LSD	CSPA	MCLA	DECLU
ECOLI	0.79	0.80	0.68	0.75	0.79	**0.82**
WINE	0.72	0.72	0.72	0.69	0.70	**0.73**
MFEAT	0.76	0.75	**0.78**	0.57	0.66	**0.78**
SEGMENT	0.62	**0.63**	0.59	0.51	0.60	**0.63**
OPTDIGITS	0.84	0.85	0.81	0.77	0.80	**0.90**
PENDIGITS	0.74	0.77	0.84	0.65	0.68	**0.85**

we have used $\sigma = 0.02$ in noise generation and $K = 5$ in K-Nearest-Neighbor similarity graph. For all cluster ensemble methods, we have used $M = 10$ base clusterings.

The clustering performance is evaluated by cluster purity, calculated by $\text{purity} = \frac{1}{N} \sum_{k=1}^{r} \max_{1 \leq l \leq r} n_k^l$, where n_k^l is the number of data samples in the cluster k that belong to ground-truth class l. A larger purity in general corresponds to a better clustering result.

The resulting cluster purities are shown in Table 2. We can see that DECLU yields top performance for all data sets, with a tie with a different method on two (`MFEAT` and `SEGMENT`) out of the six data sets in total.

5 Conclusion

In this paper we have proposed a new clustering method which consists of two steps. First, we repeatedly incorporate a small amount of noise to the data to generate multiple base partitions of the data. Next, we have developed a new ensemble method using information theoretic metric and its multiplicative optimization algorithm. Empirical studies on the new method indicate that it outperforms several other existing clustering approaches in terms of clustering accuracy.

In this work we used a fixed amount of Gaussian noise. Other types of noise would be interesting to study in the future. Similarly, it would be valuable to investigate how to automatically adjust the noise level for generating better base clusterings. Moreover, we aim at examining other information divergence (e.g., [3,20]) or mutual information variants besides D_{sum} to improve the ensemble performance (e.g., [7,12,19]). Other types of base clustering generation methods (e.g., [13,14]) could further improve the accuracy. In summary, the consistently satisfactory performance of DECLU and its computational scalability suggest considerable potential for further development of denoising based clustering methods.

Appendix: Proof of Theorem 1

Proof. We use W for current estimate, $\widetilde{W}$ for variable, and W^{new} for the new estimate, respectively. The objective function $\widetilde{\mathcal{J}}$ fulfills the theorem conditions in [16]. Therefore, we can construct the majorization function

$$G(\widetilde{W}, W) = \sum_{ik} \left[\nabla_{ik}^{+} \widetilde{W}_{ik} - \nabla_{ik}^{-} W_{ik} \log \widetilde{W}_{ik} + \frac{B_{ik}}{A_{ik}} W_{ik} - \frac{W_{ik}}{A_{ik}} \log \widetilde{W}_{ik} \right] + \text{constant}$$

such that $G(\widetilde{W}, W) \geq \widetilde{\mathcal{J}}(\widetilde{W}, \lambda)$ and $G(W, W) = \widetilde{\mathcal{J}}(W, \lambda)$. Let W^{new} be the minimum of $G(\widetilde{W}, W)$, which is implemented by zeroing $\partial G / \partial \widetilde{W}$ and yields Eq. 12. Therefore $\widetilde{\mathcal{J}}(W^{\text{new}}, \lambda) \leq G(W^{\text{new}}, W) \leq G(W, W) = \widetilde{\mathcal{J}}(W, \lambda)$.

References

1. Arora, R., Gupta, M., Kapila, A., Fazel, M.: Clustering by left-stochastic matrix factorization. In: ICML (2011)
2. Bishop, C.: Training with noise is equivalent to Tikhonov regularization. Neural Comput. **7**(1), 108–116 (1995)
3. Dikmen, O., Yang, Z., Oja, E.: Learning the information divergence. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1442–1454 (2015)
4. Herbrich, R., Graepel, T.: Invariant pattern recognition by semidefinite programming machines. In: NIPS (2004)
5. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR, pp. 50–57 (1999)
6. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33**(3), 1065–1076 (1962)
7. Romano, S., Bailey, J., Nguyen, V., Verspoor, K.: Standardized mutual information for clustering comparisons: one step further in adjustment for chance. In: ICML (2014)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
9. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
10. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. In: ICML (2008)
11. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
12. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)
13. Yang, Z., Hao, T., Dikmen, O., Chen, X., Oja, E.: Clustering by nonnegative matrix factorization using graph random walk. In: NIPS (2012)
14. Yang, Z., Laaksonen, J.: Multiplicative updates for non-negative projections. Neurocomputing **71**(1–3), 363–373 (2007)
15. Yang, Z., Oja, E.: Linear and nonlinear projective nonnegative matrix factorization. IEEE Trans. Neural Netw. **21**(5), 734–749 (2010)
16. Yang, Z., Oja, E.: Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization. IEEE Trans. Neural Netw. **22**(12), 1878–1891 (2011)
17. Yang, Z., Oja, E.: Clustering by low-rank doubly stochastic matrix decomposition. In: ICML (2012)
18. Yang, Z., Oja, E.: Quadratic nonnegative matrix factorization. Pattern Recogn. **45**(4), 1500–1510 (2012)
19. Yang, Z., Peltonen, J., Kaski, S.: Optimization equivalence of divergences improves neighbor embedding. In: ICML (2014)
20. Yang, Z., Zhang, H., Yuan, Z., Oja, E.: Kullback-leibler divergence for nonnegative matrix factorization. In: Honkela, T. (ed.) ICANN 2011, Part I. LNCS, vol. 6791, pp. 250–257. Springer, Heidelberg (2011)
21. Zhu, Z., Yang, Z., Oja, E.: Multiplicative updates for learning with stochastic matrices. In: Kämäräinen, J.-K., Koskela, M. (eds.) SCIA 2013. LNCS, vol. 7944, pp. 143–152. Springer, Heidelberg (2013)