Fast Affinity Propagation Clustering based on Machine Learning

Shailendra Kumar Shrivastava[1], Dr. J.L. Rana[2] and Dr. R.C. Jain[3]

[1] Samrat Ashok Technological Institute
Vidisha, Madhya Pradesh 464 001, India

[2] Ex Head of Department, CSE, M.A.N.I.T
Bhopal, Madhya Pradesh, India

[3] Samrat Ashok Technological Institute
Vidisha, Madhya Pradesh 464 001, India

Abstract

Affinity propagation (AP) was recently introduced as an un-supervised learning algorithm for exemplar based clustering. In this paper a novel Fast Affinity Propagation clustering Approach based on Machine Learning (FAPML) has been proposed. FAPML tries to put data points into clusters based on the history of the data points belonging to clusters in early stages. In FAPML we introduce affinity learning constant and dispersion constant which supervise the clustering process. FAPML also enforces the exemplar consistency and one of 'N' constraints. Experiments conducted on many data sets such as Olivetti faces, Mushroom, Documents summarization, Thyroid, Yeast, Wine quality Red, Balance etc. show that FAPML is up to 54 % faster than the original AP with better Net Similarity.

Keywords: *clustering, affinity propagation, exemplar, machine learning, unsupervised learning*

1. Introduction

Clustering is a fundamental task in computerized data analysis. It is concerned with the problem of partitioning a collection of data points into groups/categories using unsupervised learning techniques. Data points in groups are similar. Such groups are called clusters [1][2][3]. Affinity propagation [6] is a clustering algorithm which for given set of similarities (also denoted by affinities) between pairs of data points, partitions the data by passing the messages among the data points. Each partition is associated with a prototypical point that best describes that cluster. AP associates each data point with one such prototype. Thus, the objective of AP is to maximize the overall sum of similarities between data points and their representatives. Affinity propagation clustering algorithm is slow. Fast affinity algorithms for clustering find the clusters in less time as compared to AP. Efforts of Earlier researcher to make AP fast, yielded only limited benefits. Proposed FAPML finds the clusters in much less time as compared to AP and net similarity is much better than AP. We will first discuss the disadvantages of exiting Fast AP. FSAP [9] constructs the sparse similarity matrix by K-nearest neighbor algorithm .The FSAP does not give same result as AP. FSAP uses heuristic approach to find K. This reduces the cluster quality. FAP (based on message pruning) [10] Prunes the unnecessary messages exchange among data points in iterations to compute the convergence. This algorithm requires the extra time to find the necessary and unnecessary messages. Fast affinity propagation clustering (based on sampling of data points) [11] algorithm applies the sampling theorem to choose a small number of representative exemplar whose number is much less than data points but larger than the clusters. Clustering quality is still not as good as AP.

In the Literature[5] Machine Learning is defined as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Proposed FAPML is based on this definition of machine learning. FAPML does not have disadvantages of earlier reported fast affinity propagation algorithms. FAPML enforces the one of 'N' constraint and exemplar consistency. One of 'N' constraint means that data points belong exactly in one cluster. Exemplar consistency mean if other data points do not choose the given data point as exemplar than given data point cannot choose itself as an exemplar. Proposed FAPML tries to put data points into clusters based on the history of the data points belonging to clusters in early stages. Proposed algorithm has affinity learning constant and dispersion constant. By Affinity learning constant it uses experience in the clustering process to put data points in same clusters and by dispersion constant it uses experience in the process of clustering to put data point into different clusters.

The remainder of this paper is organized as follows. Section 2 gives a brief over view of original Affinity Propagation algorithm, FSAP, Fast algorithm for Affinity propagation (based on message pruning), Fast affinity propagation clustering (based on sampling of data points). Section 3 introduces the main idea and details of our algorithm. Section 4 discusses the experimental results and evaluation. Section 5 provides the concluding remarks and future directions.

2. Related works

Before we go into details of our FAPML approach, we would briefly review some works that are closely related to this paper. FSAP, Fast algorithm for Affinity propagation (based on message pruning) and *Fast* affinity propagation clustering (based on sampling of data points) will be discuss. For the sake of continuity affinity propagation algorithm will be discuses first.

2.1 Affinity Clustering Algorithms

Affinity clustering algorithm [6][9][10] is based on message passing among data points. Each data point receives the availability from others data points (from exemplar) and send the responsibility message to others data points (to exemplar). Sum of responsibilities and availabilities for data points identify the exemplars. After the identification of exemplar the data points are assigned to exemplar to form the clusters. Following are the steps of affinity clustering algorithms.

1. Initialize the availabilities to zero $a(i,k) = 0$
2. Update the responsibilities by following equation.
$r(i,k) \leftarrow (s(i,k) + \max_{k'st\ k' \neq k}\{a(i,k' + s(i,k')\}$ Where $s(i,k)$ is the similarity of data point i and exemplar k.
3. Update the availabilities by following equation
$$a(i,k) \leftarrow min\left\{0, r(k,k) + \sum_{i's.t.i' \notin \{i,k\}} max\{0, r(i',k)\}\right\}$$
Update self-availability by following equation
$$a(k,k) \leftarrow \sum \max(\{0, r(i',k)\})$$
4. Compute sum = $a(i,k) + r(i,k)$ for data point i and find the value of k that maximize the sum to identify the exemplars.
5. If Exemplars do not change for fixed number of iterations go to step (6) else go to Step (1)
6. Assign the data points to Exemplars on the basis of maximum similarity to find clusters.

2.2 Fast sparse affinity propagation (FSAP)

Jia et al [9] proposed fast sparse affinity propagation (FSAP) clustering algorithm. First step is construction of sparse similarity matrix. Presume that the data points that are far apart will not choose each other as an exemplar and set the similarity between them as zero. Construct the similarity matrix by K-nearest neighbor algorithm .Second step is iterative edge refinement. Data points that serve as good exemplar locally may be candidate for exemplar globally. Third step uses AP to find exemplar and clusters. Complexity of this algorithm is O(NT). Where N is number of data point and T is number of non-zero entries in sparse matrix. Jia et al apply this algorithm for organizing of image Search results obtained from state-of-the-art image search engines. It discovers exemplars from search results and simultaneously groups the images. The exemplars are delivered to the user as a summary of search results instead of the large amount of unorganized images. The FSAP does not give same result as AP. FSAP uses heuristic approach to find K. The improper value of K reduces the cluster quality.

2.3 Fast algorithm for Affinity propagation (based on message pruning)

Fujiwara et al. [10] proposed Fast algorithm for Affinity propagation (FAP). FAP overcomes the drawback of FSAP. Computational Complexity is $O(N^2 + MT)$ .Where N is number of data points; M is number of entries in similarity matrix. T is the number of iterations. FAP prunes the unnecessary message exchanges among data points in each iteration to compute the convergence (Mathematically Proved that unnecessary pruned messages can be recovered from un-pruned message). Then Computes the convergence values of pruned message from the un-pruned messages. Rest of the algorithm steps is same as AP.

2.4 Fast affinity propagation clustering (based on sampling of data points)

Shang et al. [11] proposed fast affinity propagation clustering (FAP). This Algorithm applies the fast sampling theorem to choose a small number of representative exemplar whose number is much less than data points and larger than the clusters. Secondly the representative exemplar is assigned cluster labels by a density-weighted spectral clustering method. In First step the graph is coarsened by fast sampling algorithm to collapse the

neighboring data points into subsets of representative exemplar. In second step density weighted spectral clustering is applied on set of final representative exemplars and last step is to assign cluster membership for each data point corresponding to its representative exemplar. FAP outperforms both spectral clustering and AP in terms of quality, speed, and memory usage.

2.5 Binary Variable Model for affinity propagation clustering

Givoni et al [8] proposed A "Binary Variable Model for affinity propagation clustering". It is a graphical model of AP. This model enforces two constraints. First one of 'N' Constraints $\sum_{i=1}^{N} c_{ij} = 1$ where $\{c_{ij}\}_{i=1,j=1}^{N}$ is the binary variable. One of 'N' constraint ensures that one data point belongs to exactly one exemplar/cluster. Second constraint exemplar consistency ensures that if data point k is chosen as exemplar by other data point i then k must choose itself as an exemplar. We will extend the idea of one of 'N' constraint and exemplar consistency.

3. Proposed FAPML

Fast Affinity Propagation based on machine learning takes as input a collection of real-valued similarities among data points, where the similarity $s(i,k)$ indicates how well the data point with index k is suited to be the class center for data point i. In the process of FAPML availability and responsibility messages are exchanged among data points. Initially all data points can become the candidate exemplar. The responsibility message $r(i,k)$ is sent from data point i to candidate exemplar k. The availability message (i,k), sent from candidate exemplar k to data point i. A responsibility is updated from the modified equations which are as follow.

$$r(i,k) \leftarrow (s(i,k) + al(i,k) - dl(i,k)) - \max_{k'st\ k' \neq k}\{a(i,k' + (st(i,k') + al(i,k) - dl(i,k)\} \quad (1)$$

Where $al(i,k)$ and $dl(i,k)$ are affinity learning experience and dispersion learning experience between point i and k. In this way we use the machine learning technique learning by experience. Availability message are updated by following equation

$$a(i,k) \leftarrow min\{0, r(k,k) + \sum_{i's.t.i' \notin \{i,k\}} max\{0, r(i',k)\}\} \quad (2)$$

Equation for updating self-availability

$$a(k,k) \leftarrow \sum \max(\{0, r(i',k)\}) \quad (3)$$

Next we find the exemplar for point i by finding the value of k (exemplar) that maximizes $r(i,k) + a(i,k)$. Now we enforce the one of 'N' constraint. One of 'N' constraint means each point becomes member of exactly one exemplar/cluster. This uses the array with index data point and its value is exemplar. In array only one value can be store hence each data point has exactly one exemplar.

Next we handle exemplar consistency. Exemplar consistency ensures if data point k is chosen as exemplar by other data point i then k must chose itself as an exemplar. If k does not choose itself as exemplar then assign the similarity between i and k to -∞ . This enforces the exemplar consistence. Repeat above process, if exemplar does not change fixed number iterations or changes in results are below threshold.

FAPML algorithm can be written as following.

1. Initialize the availabilities to zero $a(i,k) = 0$, initialize affinity learning variable $al(i,j) = 0$ and dispersion learning variable $dl(i,j) = 0$.
2. Update the responsibilities by following novel equation. $r(i,k) \leftarrow (s(i,k) + al(i,k) - dl(i,k)) - \max_{k'st\ k' \neq k}\{a(i,k' + (st(i,k') + al(i,k) - dl(i,k)\}$ Where $al(i,k)$ and $dl(i,k)$ are affinity learning experience and dispersion learning experience between point i and k.
3. Update the availabilities by following equation

$$a(i,k) \leftarrow min\{0, r(k,k) + \sum_{i's.t.i' \notin \{i,k\}} max\{0, r(i',k)\}\}$$

Update self-availability by following equation

$$a(k,k) \leftarrow \sum \max(\{0, r(i',k)\})$$

4. Compute sum = $a(i,k) + r(i,k)$ for data point i and find the value of k that maximizes the sum to identify the exemplars.
5. Increase $al(i,k)$ by $al\ constant$ where i and k are the index of data point and k is the index of exemplar of same cluster. Increase the value of $dl(i,k))\ by\ dl\ constant$ for data point i and exemplar k of different cluster.
6. Check exemplar chosen by other data points in step (4). If exemplar does not choose itself as an exemplar, update similarity of data points (chosen exemplar) to exemplar to minus infinity. This enforces the exemplar consistency.

7. Enforce the one of 'N' constraint.
8. If Exemplars do not change for fixed number of iterations go to step (9) else go to Step (2)
9. Assign the data points to Exemplars on the basis of maximum similarity to find clusters.

4. Experimental Results and Evaluation

In this Section, we present results and evaluation of set of experiments to verify the effectiveness and efficiency of our proposed algorithm for clustering. We conducted experiments on Olivetti faces, Mushroom, Thyroid, Yeast, document summarization, Wine quality red and balance data sets. Details of data sets are as follow:

Table 1

S.No.	Dataset	No. of Instances	Number of Attributes	References
1	Yeast	1484	8	http://archive.ics.uci.edu/ml/machine-learning-databases/yeast/
2	Olivetti faces	900	40	http://www.psi.toronto.edu/
3	Thyroid	215	6	http://archive.ics.uci.edu/ml/machine-learning-databases/Thyroid/
4	Document Summarization	125	6	http://www.psi.toronto.edu/
5	Mushroom	5807	22	http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/
6	Wine Quality Red	1599	12	http://archive.ics.uci.edu/ml/machine-learning-databases/winequality/
7	Balance	625	3	http://archive.ics.uci.edu/ml/machine-learning-databases/balance-scale/

The measures we use to compare the algorithms are the net similarity/sum of similarities of all non-exemplar data points to their exemplar and number of iterations. AP and FAPML have been run on seven data sets of table 1. Figure 1 to Figure 7 shows the variation in Net similarity with Number of iteration.

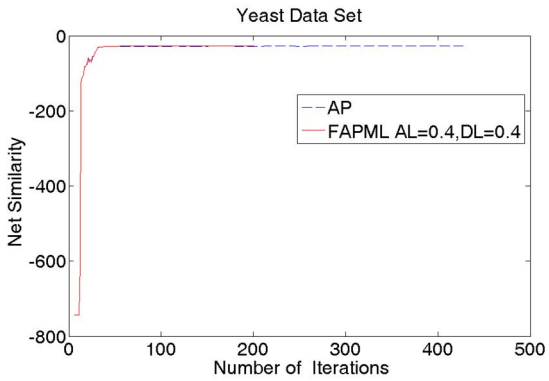

Figure 1

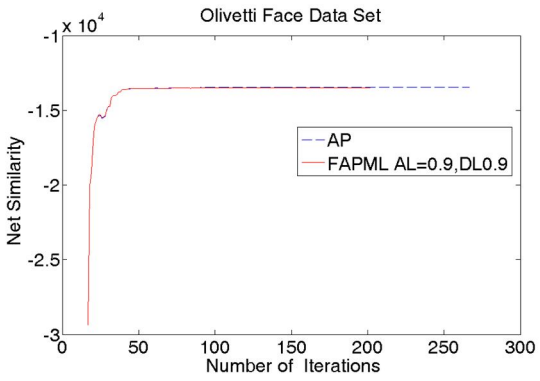

Figure 2

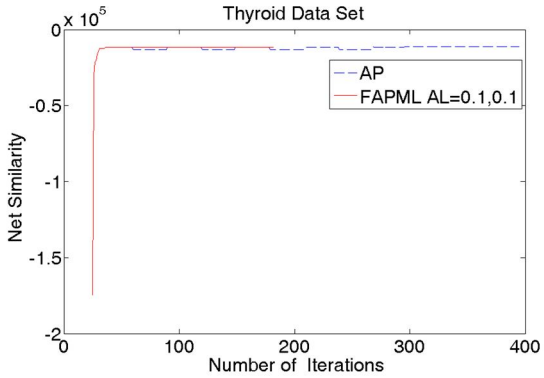

Figure 3

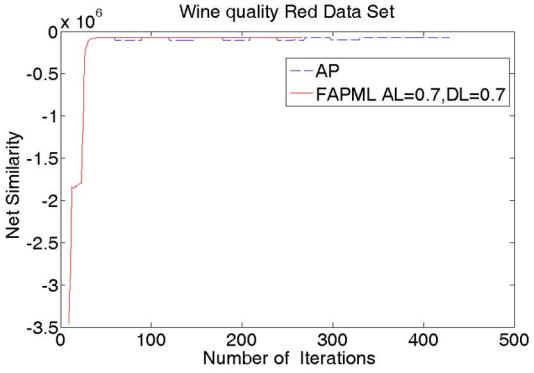

Figure 6

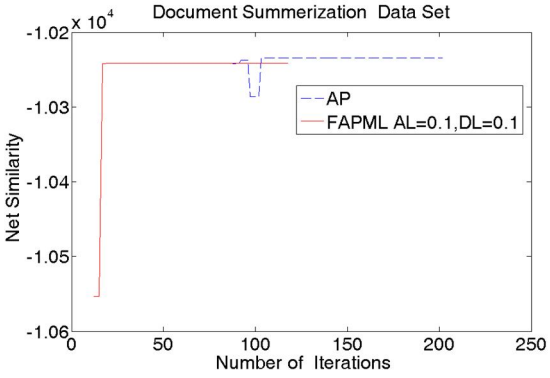

Figure 4

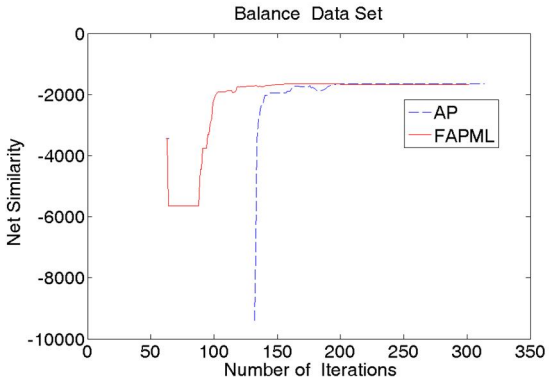

Figure 7

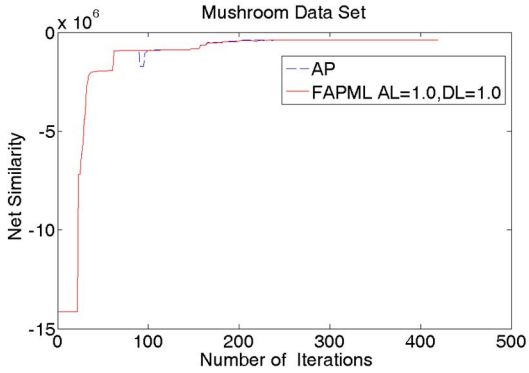

Figure 5

Figure 8 to Figure 14 shows the comparison between AP and FAPML for number of iterations and learning constants.

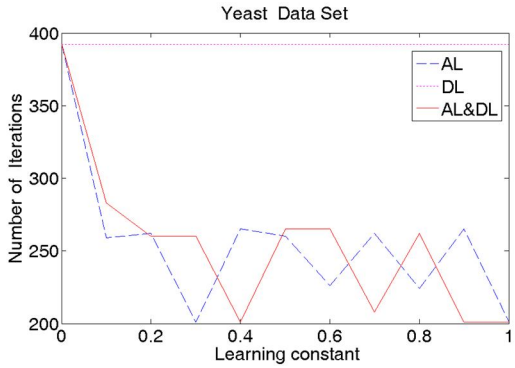

Figure 8

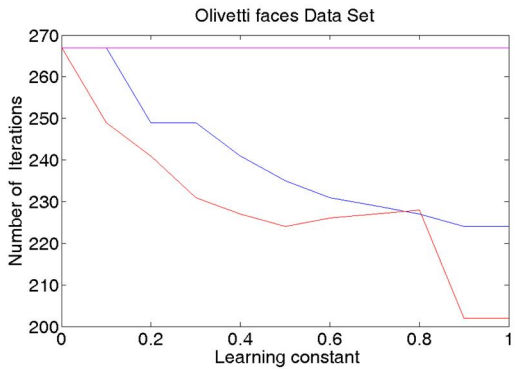

Figure 9

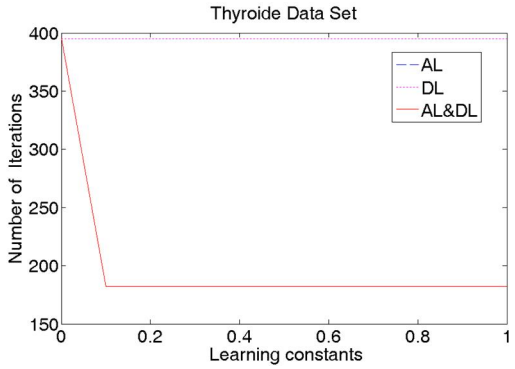

Figure10

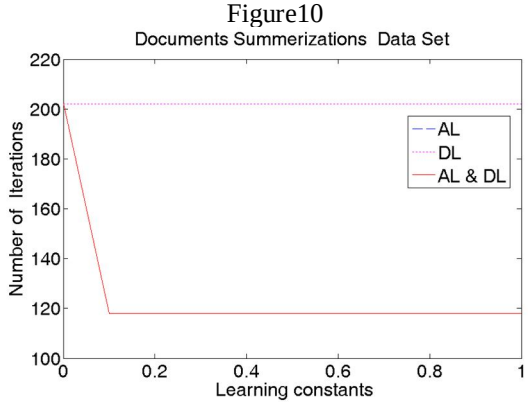

Figure 11

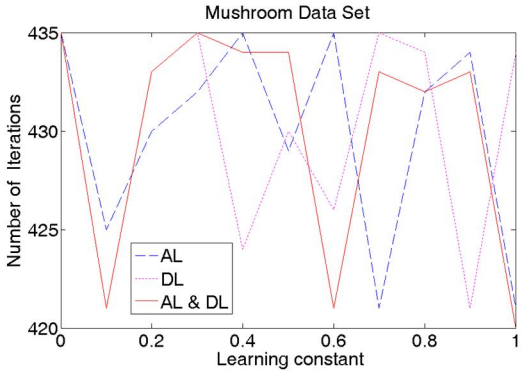

Figure 12

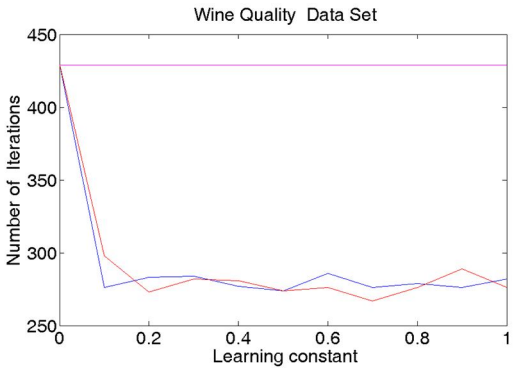

Figure 13

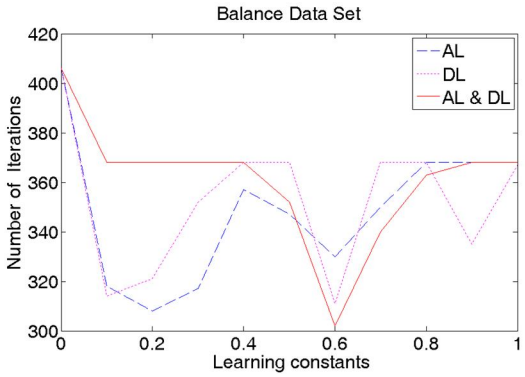

Figure 14

Following tables show the comparison between AP and FAPML.

Table 2

Name of Data Sets	Affinity Propagation			FAPML					Percentage improvement in Results
	Similarities of data points to exemplars	Number of Clusters Identified	Number of Iterations	AL Const.	DL Const.	Similarities of data points to exemplars	Number of Clusters Identified	Number of Iterations	
Yeast	-18.2992	92	392	0.4	0.4	-17.3783	104	201	48.72%
Olivetti faces	-9734.72	62	267	0.9	0.9	-9429.42	68	202	32.17%
Thyroid	-5903	14	395	0.1	0.1	-5607.66	15	182	53.92%
Document summarization	-9607.91	4	202	0.1	0.1	-9582.03	4	118	41.58%
Mushroom	-213315	126	435	1.0	1.0	-213310	126	420	3.4%
Wine Quality Red	-36516.5	36	429	0.7	0.7	-36516.5	37	267	37.76%
Balance	-1643	25	368	0.6	0.6	-1053	31	302	17.93%

Computationally the proposed FAPML algorithm outperformed AP. Net similarity achieved by proposed algorithm is also better than AP. Complexity of FAPML is $O(N^2 T)$, where N is number of data points and T is number of iterations (shown in table2 and figures 1-12). The required number of iterations T is also less than AP, which makes FAPML a Fast affinity propagation algorithm based on machine learning. Clustering quality of FAPML is also better which can be seen from Net similarity/Sum of similarities data points to exemplar. As shown in Table 2 and Figure 1-7, the Net similarity/Sum of similarities of FAPML is higher than AP. Thus the overall performance of FAPML evaluated for net similarity and time is better.

4. Concluding remarks and future directions

Recently introduced Affinity Propagation clustering is slow. In this paper we have proposed a Fast Affinity Propagation algorithm using Machine Learning, based on learning by experience principal of ML. FAPML outperforms AP in terms of speed and clustering accuracy. Extensive experiments on many standard datasets show that the proposed FAPML produces better clustering accuracy in less time.

There are a number of interesting potential avenues for future research. FAPML can be made adaptive, Hierarchical, Partitional, Incremental etc. FAPML can also be applied in Text clustering and clustering based on Heterogeneous Transfer Learning.

References

[1] RuiXu Donald C. Winch, "Clustering" , IEEE Press 2009 ,pp 1-282
[2] Jain, A. and DubesR. "Algorithms for Clustering Data ", Englewood Cliffs, NJ Prentice Hall, 1988.
[3] Jain A.K., Murthy M.N. and Flynn P.J., "Data Clustering: A Review ", ACM Computing Surveys, Vol.31. No 3, September 1999, pp 264-322.
[4] RuiXu, and Donald Wunsch," Survey of Clustering. Algorithms ", IEEE Transactions on Neural Network, Vol 16, No. 3, 2005 pp 645.
[5] EthemAlpaydin , "Introduction to Machine Learning ",Prentice Hall of India Private Limited New Dehli,2006,pp133-150.
[6] Frey, B.J. and Dueck D." Clustering by Passing Messages Between Data Points ", Science 2007, pp 972–976.

[7] Kaijun Wang, Junying Zhang, Dan Li, Xinna Zhangand Tao Guo, Adaptive Affinity Propagation Clustering",ActaAutomaticaSinica, 2007 ,1242-1246.
[8] Inmar E. Givoni and Brendan J. Frey,"A Binary Variable Model for AffinityPropagation",Journal Neural Computation,Volume 21 Issue 6, June 2009,pp1589-1600.
[9] Yangqing Jiay, Jingdong Wangz, Changshui Zhangy, Xian-Sheng Hua, "Finding Image Exemplars Using Fast Sparse Affinity Propagation", Proceedings of the 16th ACM International conference on Multimedia,2006 , pp113 – 118.
[10] Yasuhiro Fujiwara, Go Irie and Tomoe Kitahara, "Fast Algorithm for Affinity Propagation",
[11] Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp 2238-2243.
[12] Shang Fanhua,.Jiao L.C, Shi Jiarong, Wang Fei, Maoguo Gong,"Fast affinity propagation clustering: A multi-level approach",Pattern Recognition (Elseevier), 2012 ,pp 474–486.

Shailendra Kumar Shrivastava,B.E.(C.T.),M.E.(CSE) Associate Professor I.T., Samrat Ashok Technological Institute Vidisha. He has more than 23 Years Teaching Experiences. He has published more than 50 research papers in National/International conferences and Journals .His area of interest is machine learning and data mining.
Dr J.L.Rana B.E.M.E.(CSE),PhD(CSE) HE has so many publication in Journal and conferences.
Dr. R.C.Jain PhD .He is the director Samrat Ashok Technological Institute Vidisha(MP).He has published more than 150 research papers in Journals and Conferences.