Hierarchical Clustering with Proximity Metric Derived from Approximate Reflectional Symmetry

Yong Zhang* and Yun Wen Chen

Department of Computer Science and Engineering
School of Information Science and Engineering
Fudan University
Shanghai, 200433, P.R. China
zhang_yong@fudan.edu.cn

Abstract. In order to address the problems arise from predefined similarity measure, learning similarity metric from data automatically has drawn a lot of interest. This paper tries to derive the proximity metric using reflectional symmetry information of the given data set. We first detect the hyperplane with highest degree of approximate reflectional symmetry measure among all the candidate hyper-planes defined by the principal axes and the centroid of the given data set. If the symmetry is prominent, then we utilize the symmetry information acquired to derive a retorted proximity metric which will be used as the input to the Complete-Link hierarchical clustering algorithm, otherwise we cluster the data set as usual. Through some synthetic data sets, we show empirically that the proposed algorithm can handle some difficult cases that cannot be handled satisfactorily by previous methods. The potential of our method is also illustrated on some real-world data sets.

1 Introduction

Cluster analysis, as one of the basic tools for exploring the underlying structure of a given data set, has been applied in a wide variety of fields. Actually, clustering (classification) plays an important and indispensable role in the long history of human development as one of the most primitive activities[7].

As pointed by most researchers, cluster analysis intends to partition a group of objects into a number of more or less homogeneous subgroups (clusters) such that patterns within a cluster are more similar to each other than patterns belonging to different clusters. Often, a clear distinction is made between supervised clustering and unsupervised clustering, the former involving only labeled data while the latter involving only unlabeled data in the process of learning.

Existing methods for clustering fall into two categories as hierarchical clustering and partitional clustering, based on the properties of clusters generated.

* Corresponding author.

L. Wang et al. (Eds.): FSKD 2006, LNAI 4223, pp. 69–78, 2006.

Hierarchical clustering groups data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa, while partitional clustering directly divides data objects into some pre-specified number of clusters without the hierarchical structure.

Regardless of hierarchical or partitional clustering, they both rely on the definition of similarity (or dissimilarity) measure, which establishes a rule for assigning patterns to a particular cluster. Many algorithms adopt a predefined similarity measure based on some particular assumptions. These algorithms may fail to model the similarity correctly when the data distribution does not follow assumed scheme. Instead of choosing the similarity metric manually, a promising solution is to learn the metric from data automatically.

Usually, in order to extract appropriate metric from data, some additional background knowledge or supervisory information should be made available for unsupervised clustering. Supervisory information is often provided in the form of partial labeled data or pairwise similarity and/or dissimilarity constraints[4][5].

Despite the progress in semi-supervised clustering for similarity metric learning, metric learning for strict unsupervised clustering remains a challenge. An interesting trial is to extract similarity metric based on symmetry. In [6], for instance, a novel nonmetric distance measure based on the idea of point symmetry is proposed, where the point refers to the centroid of corresponding cluster.

Following a similar consideration, in this paper, we exploit the approximate reflectional symmetry of the given data set to induce the proximity metric such that the similarity between pairs of points is not strictly depend on their closeness in the feature space, but rather on their symmetrical affinity. Although this is similar to [6] to some extent, several important differences should be noticed. First, we don't assume compulsory symmetry exist in data set, we perform the symmetry detection as a preprocessing. If symmetry exist, we continue the clustering with the guidance of symmetry information detected, otherwise we process the data set as usual. Second, we consider reflectional symmetry, which is more common in abstract or in nature, rather than the point symmetry. Third, we mainly utilize the symmetry information acquired to deduce an appropriate proximity metric.

The remainder of this paper is organized as follows: Section 2 introduces the algorithm for reflectional symmetry measurement and detection. The proposed clustering algorithm, which can be divided into two stages: proximity metric construction and complete-link clustering, is described in section 3. Simulation results on both synthetic and real-world data are presented in section 4, comparing with some previous methods. In section 5 we review the related research briefly, and some concluding remarks are given in section 6.

2 Reflectional Symmetry Measurement and Detection

Symmetry is often described with symmetric transformation T, which is a transformation that when applied to all elements of a system S result in a system S' that is identical to the original system S.

Before we incorporate symmetry information into clustering, we must solve the following problem: How can we detect or measure the symmetry of the given system efficiently? If we only consider symmetry as a binary feature (i.e., a system is either symmetric or it is not symmetric), or exact symmetry, then the problem will be easier. However, even perfectly symmetric objects may lose their exact symmetry because of digitalization or quantification. Consider symmetry as an approximate feature[3], we can describe inexact symmetry more exactly. A system has approximate symmetry with respect to a transformation if it is "almost" invariant under that transformation. Obviously, the challenge is to interpret "almost" in an appropriate manner. Now, the problem can be presented as follows: How can we measure and/or detect approximate symmetry degree for a given system?

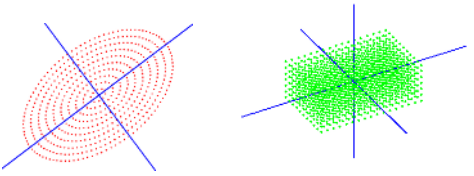
As in[3], this problem can be translated into two subproblems: First, how to measure the symmetry degree of a given data set with respect to any specified hyperplane, and second, how to find a hyperplane with highest degree of symmetry measure.

The reflectional symmetry about a hyperplane for a set of n-dimensional data points can be measured by reflecting each point through the hyperplane, and measuring the distance from the reflected point to the closest point in the original data set[3]. More accurately, the mean of these distances is a measure of reflectional asymmetry. In order to make the measure invariant to scale, we divide the measure by a scaling factor. We select the scaling factor as the maximum distance of any point from the centroid so that the asymmetry measure will be within the range zero to one for any hyperplane that contain the centroid. To create a symmetry measure from the asymmetry measure, the asymmetry measure can be subtracted from one. For perfectly symmetric data set, the symmetry measure will be one. When the degree of symmetry decreases, the symmetry measure also decreases.

Finding the most reflectional symmetric hyperplane for an n-dimensional point set requires the symmetry to be measured for every possible hyperplane. This is not practical because the number of possible hyper-planes is infinite. Even though we can narrow the search space by discretization, this approach is also computationally prohibitive especially for high dimension situations. A more practical approach is to utilize the principal axes of the data set to define n candidate hyper-planes, and choose the hyperplane associated with the highest symmetry measure as the approximate hyperplane of reflectional symmetry. The motivation for this approach is based on two theorems found in [2], which state that:

- Any plane of symmetry of a body is perpendicular to a principal axis
- Any axis of symmetry of a body is a principal axis

Fig. 1 shows the principal axes that would be found using the centroid and eigenvectors of the covariance matrix for 2D and 3D objects. A hyperplane can be uniquely defined using a point and a normal vector, so each principal axis can be used to define a hyperplane containing the centroid of the given data set.



(a) (b)

Fig. 1. The centroids and principal axes for a 2D ellipse (a) and 3D box (b)

The detailed algorithm description is as follows:

Input:
the data set $S = \{\overrightarrow{x_i}\}_{i=1}^{N}$, where $\overrightarrow{x_i}$ is represented as n-dimensional column vector

Output:
the hyperplane *symhp* with highest value of reflectional symmetry measurement, and the corresponding symmetry measure value *symv*

Steps:

Step 1 determine all the n candidate hyper-planes

1.a compute the centroid $\overrightarrow{m}$ of S

$$\overrightarrow{m} = \frac{1}{|S|}\sum_{\overrightarrow{x}\in S}\overrightarrow{x}$$

1.b estimate the covariance matrix C for S

$$C = \frac{1}{|S|}\sum_{\overrightarrow{x}\in S}(\overrightarrow{x}-\overrightarrow{m})(\overrightarrow{x}-\overrightarrow{m})^T$$

1.c compute all the n principal axis vectors $\overrightarrow{w_1}$, $\overrightarrow{w_2}$, $\cdots$, $\overrightarrow{w_n}$ based on C

1.d determine n candidate hyper-planes $\Pi_1, \Pi_2, \cdots, \Pi_n$ as:

$$\Pi_i : \overrightarrow{w_i}\cdot(\overrightarrow{x}-\overrightarrow{m}) = 0, \text{ where } \overrightarrow{x} \in R^n, i = 1, 2, \cdots, n$$

Step 2 measure the symmetry degree w.r.t.every candidate hyperplane

2.a compute $d_{max} = max_{\overrightarrow{x}\in S} \parallel \overrightarrow{x} - \overrightarrow{m} \parallel$

2.b for each candidate hyperplane $\Pi_i, i = 1, 2, \cdots, n$

2.b.1 for each $\overrightarrow{x} \in S$, compute
$d(\overrightarrow{x}, \Pi_i) = min_{\overrightarrow{y}\in S} \parallel \overrightarrow{x^*} - \overrightarrow{y} \parallel$, where $\overrightarrow{x^*}$ is the reflection of $\overrightarrow{x}$ through Π_i

2.b.2 compute $\bar{d}(\Pi_i) = \frac{1}{|S|}\sum_{\overrightarrow{x}\in S} d(\overrightarrow{x}, \Pi_i)$

2.b.3 compute $symv(\Pi_i) = 1 - \frac{\bar{d}(\Pi_i)}{d_{max}}$

Step 3 find $\Pi_k (1 \le k \le n)$, that
$symv(\Pi_k) = max_{1\le i\le n} symv(\Pi_i)$,
and return Π_k as well as $symv(\Pi_k)$

The order of complexity of this algorithm is $O(nN^2)$, where N represents the number of points in the given data set and n is the dimension of data points. However, if the data are sorted along an axis, then the typical order of complexity can be reduced to $O(nNlogN)$ by using a binary search to find the nearest neighbor. If we want to find better symmetric hyperplane, we can also choose the principal axis as the initial candidate and optimize it further using some appropriate optimization methods[3].

3 The Proposed Clustering Algorithm

3.1 Proximity Metric Construction

We have acquired approximate reflectional symmetry information about the given data set using the algorithm presented in section 2. Then how can we incorporate this information into cluster analysis so as to improve the clustering results? The direct method is to induce approximate proximity metric based on the symmetry information available so that the similarity between pairs of points is not strictly depend on their closeness in the feature space, but rather on their symmetrical affinity.

Nevertheless, just as mentioned earlier, we cannot expect reflectional symmetry present for every data set. In the proposed algorithm, we introduce a pre-specified threshold ts. If the symmetry measure with respect to the acquired hyperplane is bigger than ts, then we modify the original proximity metric based on the symmetry information before clustering, otherwise we cluster the patterns directly as usual without any proximity modification.

If the symmetry information is available, we construct Approximate Reflection Pair Set (ARPS) with respect to the symmetric hyperplane. Concretely, let $\overrightarrow{x}$, $\overrightarrow{y}$ be two patterns in S, we will add $\overrightarrow{x}$ and $\overrightarrow{y}$ into ARPS if $\overrightarrow{y}$ is the nearest neighbor of $\overrightarrow{x^*}$, where $\overrightarrow{x^*}$ is the reflection of $\overrightarrow{x}$ through the specified hyperplane. If no extra constraint considered, all the patterns in S will be added into ARPS ultimately unless the cardinal of S is odd. This may be unreasonable, because the nearest neighbor of $\overrightarrow{x^*}$ may be very far apart from $\overrightarrow{x^*}$ especially when most of the patterns in S have been added into ARPS. In order to address this problem, we only consider a pattern within the specified neighborhood of $\overrightarrow{x^*}$, i.e., the distance between $\overrightarrow{x^*}$ and its nearest neighbor is smaller than a specified threshold tn, as the approximate reflection of $\overrightarrow{x}$, otherwise $\overrightarrow{x}$ is not considered.

Then we derive a new proximity metric on the basis of the ARPS by lowering the distances between the approximate reflectional pairs, i.e., pairs in ARPS which are mutual reflections, to zero. In this way, however, the points only satisfy the symmetric affinity imposed by ARPS, instead of the reflectional symmetry of the given data set as a whole and intrinsic feature. Hence, as in [5], we also require the points to satisfy the implied symmetric relation of ARPS. We interpret the proximity matrix as weights for a complete graph over the data points. Thus we obtain a new metric by running an all-pairs-shortest-paths algorithm on the modified proximity matrix. The concrete algorithm is presented as follows:

Input:

the data set $S = \{\overrightarrow{x_i}\}_{i=1}^{N}$, where $\overrightarrow{x_i}$ is represented as n-dimensional column vector

the symmetric hyperplane *symhp* and corresponding symmetry measure *symv*

the threshold ts and tn

Output:

the proximity matrix P

Steps:

Step 1 construct $P_{N\times N}$, where $P_{ij} = \| \overrightarrow{x_i} - \overrightarrow{x_j} \|, 1 \le i, j \le N$

Step 2 if $symv < ts$, goto step 5
Step 3 ARPS construction
3.a initialize $S^{'} = S$, $ARPS = \phi$
3.b repeat until $S^{'} = \phi$
3.b.1 select $\overrightarrow{x}$ from $S^{'}$, and find $\overrightarrow{y_0}$, that
$$\parallel \overrightarrow{x} - \overrightarrow{y_0} \parallel = min_{\overrightarrow{y} \in S^{'} - \{\overrightarrow{x}\}} \parallel \overrightarrow{x^*} - \overrightarrow{y} \parallel,$$
where $\overrightarrow{x^*}$ is the reflection of $\overrightarrow{x}$ through $symhp$
3.b.2 if $\parallel \overrightarrow{x} - \overrightarrow{y_0} \parallel < tn$, add $\overrightarrow{x}$ and $\overrightarrow{y_0}$ into $ARPS$, and let $S^{'} = S^{'} - \{\overrightarrow{x}, \overrightarrow{y_0}\}$, otherwise discard $\overrightarrow{x}$ from $S^{'}$, i.e., let $S^{'} = S^{'} - \{\overrightarrow{x}\}$
Step 4 proximity adaptation
4.a set $I = \{i | \overrightarrow{x_i} \in ARPS\}$
4.b for $k \in I$, for $i = 1$ to N, for $j = 1$ to N
$P_{ij} = min\{P_{ij}, P_{ik} + P_{kj}\}$
Step 5 return P matrix

3.2 Compete-Link Hierarchical Clustering

In this paper, we use complete-link hierarchical agglomerative clustering as the clustering approach. As a matter of fact, we can use any clustering algorithm provided that its input is a proximity matrix.

As a popular hierarchical clustering algorithm, complete-link agglomerative clustering is very familiar to most researchers. For the convenience, the algorithm is presented as follows[5]:

Input:
the proximity matrix $P_{N \times N}$
Output:
the linkage $link$
Steps:
Step 1 initialize $Clusters = \{c_i$ for each pattern $\overrightarrow{x} \in S\}$, $link = \phi$, distances $d(c_i, c_j) = P_{ij}, 1 \leq i, j \leq N$
Step 2 repeat until $|Clusters| = 1$
2.a choose closest $(c_1, c_2) = argmin_{c_i, c_j \in Clusters} d(c_i, c_j)$
2.b add (c_1, c_2) to $link$
2.c merge c_1 and c_2 into c_{new} in $Clusters$
2.d for $c_i \in Clusters$
$d(c_i, c_{new}) = max\{d(c_i, c_1), d(c_i, c_2)\}$

3.3 Algorithm Analysis

The proposed algorithm requires $O(nN^2)$ computations and $O(N^2)$ space which may be intractable for some large data sets. Notice that we allow a many-to-many correspondence for data points when we measure the degree of symmetry with respect to a specified hyperplane (section 2). However, for the construction of ARPS, we impose a one-to-one correspondence, i.e., each reflected point is only to find its nearest neighbor among the points that have not previously

been matched. The reason is that we want to find more accurate reflectional symmetric pairs to describe the symmetry information.

4 Simulation Results

4.1 Evaluation Criteria

Rand Index[7] can be used to reflect the agreement of two clustering results. Let n_s, n_d be the number of point pairs that are assigned to the same/different cluster(s) in both partitions respectively. The Rand Index is defined as the ratio of (n_s+n_d) to the total number of point pairs, i.e., $N(N-1)/2$, where N represents the number of points in the given data set. The Rand Index lies between 0 and 1, and when the two partitions are consistent completely, the Rand Index will be 1.

In what follows, we use Rand Index to measure the agreement between the resultant partition and the ground truth.

4.2 Simulation with Synthetic Data

The synthetic data is designed to highlight the improvements brought by the proposed algorithm compared with other clustering algorithms like single-link, complete-link and SBKM[6]. The parameter ts and tn are chosen for 0.8 and 0.5, respectively, and this is kept the same irrespective of the data sets used. Fig.2 shows the target clustering and the corresponding clustering results generated by the proposed algorithm and other three algorithms.

Table 1 shows the degree of reflectional symmetry for every synthetic data set and the corresponding Rand Index values for the proposed algorithm as well as other three algorithms: single-link, complete-link, and SBKM.

Table 1. The reflectional symmetry degree of the three synthetic data sets and the Rand Index values for the proposed algorithm as well as other three algorithms: SL (Single-Link), CL (Complete-Link), SBKM (Symmetry Based K-Means) after applying to the three sets

Data Sets	Data set1	Data set2	Data set3
Symmetry Degree	0.9328	0.9171	0.9458
CL	0.5896	0.4982	0.5482
SL	**1.00**	0.7207	0.5619
SBKM	0.4962	0.5246	0.5118
The Proposed Algorithm	**1.00**	**1.00**	**0.9867**

Compared with other three algorithms, the proposed algorithm improves the clustering results substantially. Although single-link clustering can also detect some of these patterns, we should notice that: with the guidance of symmetry information, both single-link and complete-link algorithms are effective for these

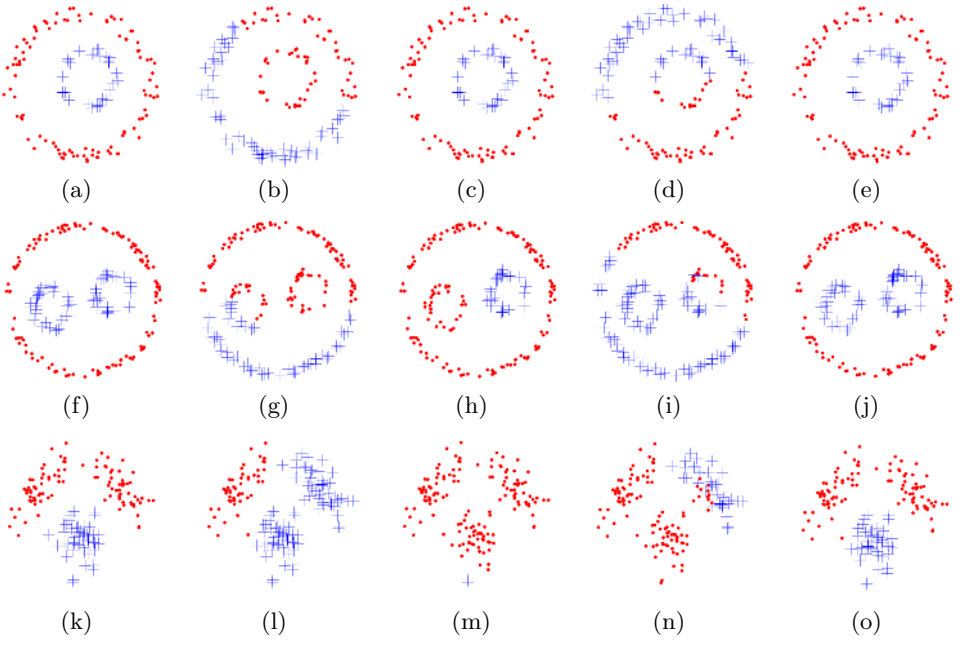


Fig. 2. Comparison of the proposed algorithm with Single-link, Complete-link, SBKM on four synthetic data sets. Subfigures in the first column show the target clusterings. Subfigures in the second, third, fourth and fifth column are the clustering results after applying Complete-link, Single-link, SBKM and the proposed algorithm, respectively.

data sets. This means the algorithm choice may be of less importance when symmetry information is available. Taken in this sense, we can indeed expect to improve the clustering results using acquired symmetry information even for a poorly-performing algorithm.

4.3 Simulation With Real-World Data

Simulations were conducted on three data sets from UCI repository: GLASS, PIMA, and BUPA[1]. Table 2 summarizes the properties of the three data sets: the number of instances, the number of dimensions (attributes), the number of classes as well as the symmetry degree acquired with the algorithm presented in Section 2.

Table 2 also shows the corresponding Rand Index values for the four algorithms considered. During the simulations, for SBKM, the parameter θ was chosen as 0.27, 0.3 and 0.18 for the three data sets respectively. And the results was averaged over 10 runs. As for the proposed algorithm, we set the parameter ts to 0.8 for all of the data sets to be studied, while the parameter tn was chosen as 0.16, 0.6 and 0.43 for the three data sets respectively.

Although the performance improvement compared with other three algorithms are not so substantial, the results are also encouraging since this indicates

Table 2. The Rand Index values for the proposed algorithm as well as other three algorithms: SL(Single-Link), CL(Complete-Link), SBKM(Symmetry Based K-Means) on the real-world data sets. The corresponding symmetry degrees(SD) are also presented. TPA is short for The Proposed Algorithm.

Dataset	#Classes	#Instances	#Attributes	#SD	CL	SL	SBKM	TPA
GLASS	7	214	9	0.9998	0.5822	0.2970	**0.6697**	0.6313
PIMA	2	768	8	0.9127	0.5466	0.5458	0.5185	**0.5559**
BUPA	2	345	6	0.9412	0.5056	**0.5104**	0.5026	**0.5104**

that our algorithm is not only confined to low-dimensional and two-category problems.

From the simulation results over synthetic and real-world data, we can confirm the effectiveness of the proposed algorithm. Different from previous algorithms, the proposed algorithm intends to learn the proximity metric using the symmetry information of the given data set. As we know, if reflectional symmetry is present in a candidate data set, then the symmetry should be considered a more significant feature or constraint. Hence, we can expect better results for the proposed algorithm when applied to data sets with approximate symmetry.

5 Related Work

Leaning metric from data has been an active research field. One traditional important family of algorithms that (implicitly) learn metrics is the unsupervised ones that take an input data set, find and embedding of it in some space. This includes algorithms such as Multidimensional Scaling (MDS) and Locally Linear Embedding (LLE) etc.

In the context of clustering, a promising approach was proposed by Wagstaff et al[8] for clustering with constraint (similarity) information. Pablo et al[4] also exploited some pairs of points with known dissimilarity value to teach a dissimilarity relation to a feed-forward neural network. Eric.P et al[9] learned a distance metric based on given examples of similar pairs of points in feature space.

The proposed algorithm is distinct from all the algorithms mentioned above. It exploits the reflectional symmetry information detected from data to derive an appropriate proximity metric over the input space. The motivation for the proposed algorithm is in accordance with the consideration in [6]to some extent.

Symmetry detection and measurement is also a key problem for the proposed algorithm. There are many literatures available about symmetry detection and/or measurement. In [2][3], the principal axes of the given data set were used as the initial setting for finding the symmetric plane(hyperplane) further. In this paper, however, we directly exploit the principal axes to determine the symmetric hyper-planes.

6 Conclusion

A method to derive the proximity metric based on the approximate reflectional symmetry information acquired from data has been proposed for complete-link hierarchical clustering. The results over several synthetic data sets demonstrate that the proposed algorithm can improve the clustering accuracy compared with other similar algorithms: single-link, complete-link and SBKM algorithm.

In this paper, we only consider to exploit the reflectional symmetry. In future, we intend to guide cluster analysis with more kinds of symmetry such as rotation, skew symmetry etc. Additionally, we are interested in extending the global symmetry detection to local symmetry detection.

Acknowledgments

This work is supported by Natural Science Foundation of China under grant No.60305002 and China/Ireland Science and Technology Collaboration Research Fund under grant No.CI-2004-09.

References

1. Blake, L., Merz, J.: UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html` (1998)
2. Changming Sun., Sherrah, J.: 3D symmetry detection using the extended Gaussian image. Pattern Analysis and Machine Intelligence, IEEE Transactions on. **19(2)** (1997) 164–168
3. Colliot, O., Tuzikov, A., Cesar, R., Bloch, I.: Approximate reflectional symmetries of fuzzy objects with an application in model-based object recognition. Fuzzy Set and Systems. **147** (2004) 141–163
4. Corsini, P., Lazzerini, B., Marcelloni, F.: A fuzzy relational clustering algorithm based on a dissimilarity measure extracted from data. Systems, Man and Cybernetics, Part B, IEEE Transactions on. **34(1)** (2004) 775–781
5. Klein, O., Kamvar, S.O., Manning, C.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. Proceedings of the Nineteenth International Conference on Machine Learning. (2002) 307–314
6. Muchun Su., Chien-Hsing Chou.: A modified version of the k-means algorithm with a distance based on cluster symmetry. Pattern Analysis and Machine Intelligence, IEEE Transactions on. **23(6)** (2001) 674–680
7. Rui Xu., Wunsch, D.: Survey of clustering algorithms. Neural Networks, IEEE Transactions on. **16(3)** (2005) 645–678
8. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. Proceedings of the Eighteenth International Conference on Machine Learning. (2001) 577–584
9. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. Advances in Neural Information Processing Systems **15** (2003) 505–512