

4 Real World Application

4.1 Financial Data Mining

Finding investment reference information from historical data is a common used strategy for investment guidance. Based on the investment reference information, similarity measurement on market behavior from different time periods are often conducted. Then, these similarity measurements can be used to retrieve useful investment information. Accordingly, an investor may decide when will be the right time to buy or to sell a targeted stock.

Also, investment experts look for stocks of similar behavior patterns in specific time periods, to make investment suggestions to help people making their investment decisions. However, there are too many data records in financial archives to be used as investment reference information, automated search mechanisms are needed to mining meaningful and useful stock behavior patterns from financial archives.

In the past, numerous researches[6] are proposed to model the dynamic behavior based on a single financial series. Often, they estimated the possible dependencies between past and future data according to various assumptions and data characteristics, such as linearity, mutual dependency, and so on. However, instead of using the estimated *instantaneous* dynamics as the description of stock market behavior, the *estimated dynamics* are used to perform short term predictions. That is, people mainly focus on using the time dependency of financial series to model system dynamics.

Recently, Terasvirta et al. [27] present an excellent survey of linear model, auto-regressions, and neural networks for forecasting macroeconomics time series. By learning the behavior of economy systems from historical data, these models are used to predict the value changes in the near future. For

instance, Pao[26] proposed an artificial neural network(ANN) to predict the electricity price using direct forecasting approach based on historical data. Lin and Chen[18] proposed a method which integrates the best features of several classification approaches by genetic-based hybrid approach, to predict the possibility of corporate failure.

On the other hand, several researchers proposed to estimate the probability of a business-cycle. Gregoir and Lengart[10] used business survey data and HMM model to find turning points of business cycles. They also found that multivariate data can be used to detect turning points of business cycles. Bellone and Gautier[4] also used Gregoir's model in a set of four financial series to achieve an unrevised and reliable advanced qualitative probabilistic indicators. However, these models are proposed to capture the dependencies between input and output data as detailed as possible and are not intended to achieve reasoning from the learned parameters. In other words, one may have probabilistic information of certain stock market activities, but acquires little or nothing about what is real happening.

More reasoning information is always desired, so that basic knowledge and the estimated results from financial archive can be combined to conclude useful investment suggestions. By computing the similarities among records sets in financial archives by the proposed Polygon descriptor and Deforming distance method, the data set in financial archives can therefore be searched, clustered, and so on. Since the Polygon descriptor reflects the system behavior, the measured similarity values reflects the change of system behavior.

Figure IV.20 shows two data distributions of stock price and EPS in 1996 and 2006 respectively. According to the shape difference between these two data distributions, we can see that the stock market behavior in 1996 and

2006 are quite different. Thus, by measuring the similarities between system behavior in different months, the change point of system behavior can be located too.

Since, Taiwan Stock market had a clear turning point around 2000, data collected during the period from 1998 to 2006 are selected as benchmark. As shown in Figure IV.21, Taiwan stock market price index dropped since 2000 and reach its lowest point before 2002. Then the index raising and falling down again until 2003. After 2003, the stock index shows an clear trend of going up. Therefore, we are targeting to find a method which can correctly locate the turning point at 2003 and partition the stage of market behavior.

A prototype system is designed to achieve the following goals. 1) Unlike most financial archives which simply list data series, the proposed method provides query by example to help users to find investment reference information. That is, by grouping data records into data sets, desired data sets can be queried by specific data set. For example, user can use all data records in a month to query for monthly similar data dependencies. 2) Unlike most model-based prediction systems which forecast the future values based on a **blackbox**, the shape of data distribution of related data sets can be used to explain the results.

4.2 System Implementation

The proposed system contains three parts: 1) A Polygon descriptor, 2) A deforming measurement method for Polygon descriptor, and 3) Web-based user interfaces. By using polygon descriptors to depict the shape of a data distribution, the dependencies among stock market quantities, such as stock price and EPS (earn per share), are extracted and characterized. Based on a set of deforming operators, the similarity between two polygon descriptors

can be measured. By using the measured similarity values, web-based user interfaces can be built to help user realize how the market behavior changes, and inspect financial archives or related news according to the similarity index.

At first, stock price and earn per share (EPS) data are collected from Taiwan Stock Market during the period from 1986 to 2006[2]. Then, monthly data are used to draw data distribution in each month. According to these data distributions, polygon descriptors are estimated to represent the system behavior in every months. By measuring the similarity values between polygon descriptors, similarities between system behavior in different months can be estimated and used to compose a similarity table which show how a system changes its behavior. Last, web-based interfaces are prototyped and demonstrated based on the estimated similarity values.

Polygon Description

First, 243 months of stock price and EPS data are used to estimate a total of 243 polygon descriptors. Thereafter, the 243 polygon descriptors are called 243PD set. The estimated Polygon descriptor for 2003/February is illustrated in Figure IV.22. The input data points are clustered into 4 groups based on the estimated 4 normal vectors of the estimated Polygon descriptor. As shown in this example, a Polygon contour in thick lines and 4 normal vectors are used to represent the input data distribution.

Similarity Table

Then, the similarity values between every two polygon descriptors in the 243PD Set, are measured by using the proposed deforming distance method. Figure IV.23 depicts the measured similarity values between every two month

data. The grey intensity of each pixel (x, y) in Figure IV.23 represents the similarity between the x^{th} month and the y^{th} month during the period from January 1986 to March 2006.

As shown in Figure IV.23, the monthly data are similar to each other during the period between 2003 and 2006. Also, the monthly data in 199X are quiet similar to each others too. However, data in these two periods are not similar to each other. By observing the shape of stock data distribution, we noticed that the data points in the period from 2003 to 2006 are distributed more or less along a line. That means the stock price in the period from 2003 to 2006 is likely dependent on EPS, and the stock price in 90s is unlikely dependent on EPS. Figure IV.22 illustrates the data distribution of the period in February 2003. That is, the change of data distribution reflects that the investors of TW stock market start to pay attention to the *intrinsic value* of stock after the crash at 2002.

Generally, three patterns are recognized at Taiwan Stock Market. The first pattern happened before Asia Financial Crisis(1997-1998). And the second pattern can be found during the period between Asia Financial Crisis and the Subprime Lending Crisis in 2007. The third pattern is recognized after 2007. However, force all periods into three patterns ignores too many detailed on the change of system behavior. That is, financial crisis, such Asia Financial Crisis and Second Lending Crisis, may effect the market for several years. Various small patterns can be found before the market settle down. Besides, the correspondence of stock market is usually late than events, such as the indication of financial crisis. Whatever, the proposed methods partition the stock market into periods (the white blocks in similarity matrix). And , the transition periods (dark bands in similarity matrix) can be found by the proposed methods too. Since the last year of collected data is before

2007, the third period, in corresponding to the third pattern, can not be observed. There are still two static periods can be found from the similarity matrix. However, the boundary between the first and the second stage in corresponding to the first and the second pattern is shifted to 2000. Actually, Asia Financial Crisis didn't affects Taiwan Stock Market very much. I believe that the divide between the first and the second stage is because Dot-com Bubble ended in 2000, and Taiwan stock market need several years to settle down to another balance.

Web-based User Interface

Based on the similarity measurement of the proposed method, a prototype system is established for public trial, evaluation, comments and suggestions. The home page of the prototype system is shown in Figure IV.24. The contents of the home page are introduction and user instructions of the prototype system. The main user interface is shown in Figure IV.25. By selecting a reference month at the left-top corner, the similarity value between the reference month and the months during the period from 1986 to 2006 are drawn. The scale of the similarity drawing can be adjusted from the right-top panel. Then a user can select their interested month by clicking the similarity drawing. The data distribution of reference month and selected month are also shown below. Besides, the data sheet of selected month can be browsed by click the button at left-bottom corner.

Figure IV.26 shows a data display window. A user selects data items to be browsed first, then click the **Query** button to retrieve the associated data items.

As shown in Figure IV.27, clicking the button at the right-bottom corner of the main user interface sends a query to Google News Archive to retrieve

the related news of the selected month.

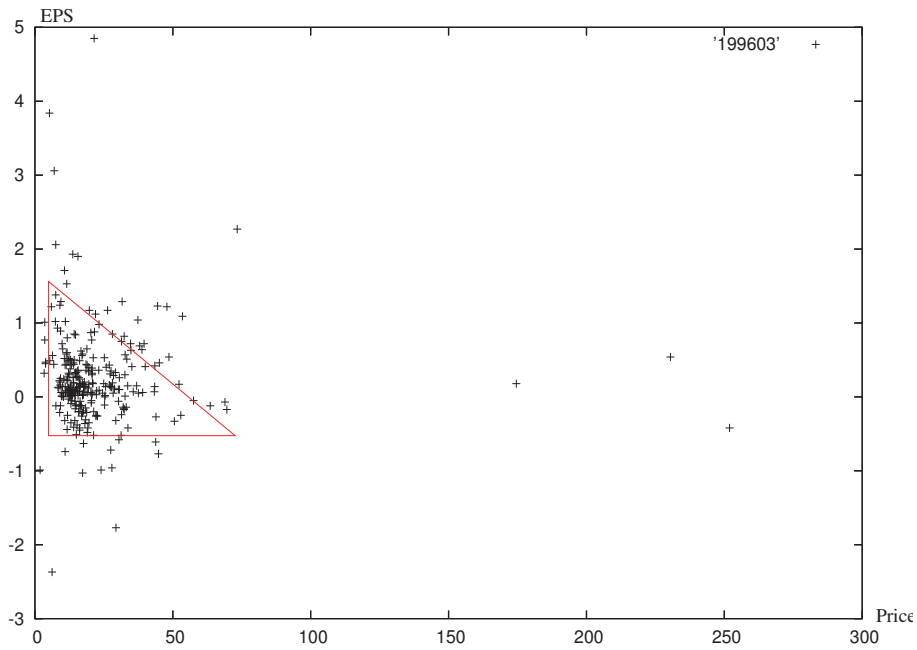
5 Concluding Remarks

Polygon descriptor (see Chapter III) models the shape of input data distribution by a reference center and normal vectors. For each Polygon descriptor, a number sequence can be measured based on the angle between adjacent normal vectors. By measuring the similarity between two number sequences using the proposed **Deforming distance** method, a similarity value for these two data distributions is available. Thus, the proposed methods can be used to build the similarity index among data sets.

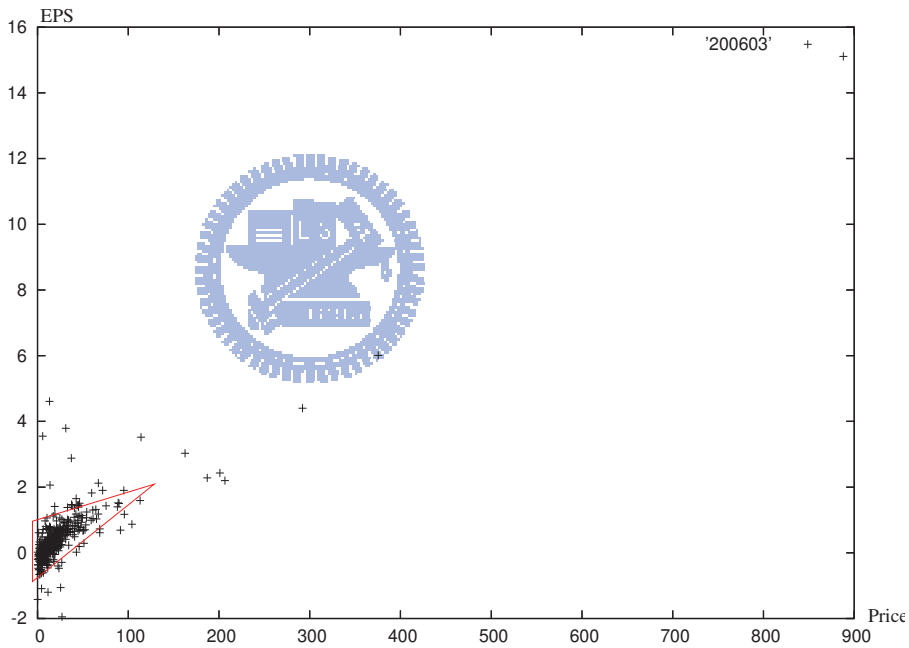
In addition, by creating new deforming operations or removing existed ones, the definition of similarity can be modified to fulfill the requirements of special applications. For example, by creating a rotation operation with zero cost, the similarity measurement can be variant to rotation. However, to measure the deforming distance, the angle magnitude between adjacent normal vectors has to be estimated first. Therefore, implementing the deforming distance measurement method is difficult for higher dimension data.

Thus, another similarity measurement method for Polygon descriptor in arbitrary dimension is proposed. *Minimal-cost normal vector match* method finds a minimal-cost multiple-to-multiple onto match between two polygon descriptors and represents the similarity between two polygon descriptors according to the cost of multiple-to-multiple onto match.

The demonstrated stock market application shows that the deforming of data distribution between EPS and stock price, meets the markets change, which we already known. If the basic assumption, that the behavior of investment decides the movement of market, holds, we can always get hints about in which state current market is.



(a) March 1996



(b) March 2006

Figure IV.20: Data distribution of stock price V.S EPS (earn per share) in 1996 (a) and 2006 (b). From the shape of these two distributions, the data distribution between stock price and EPS data in 1996 and 2006 are quite different.

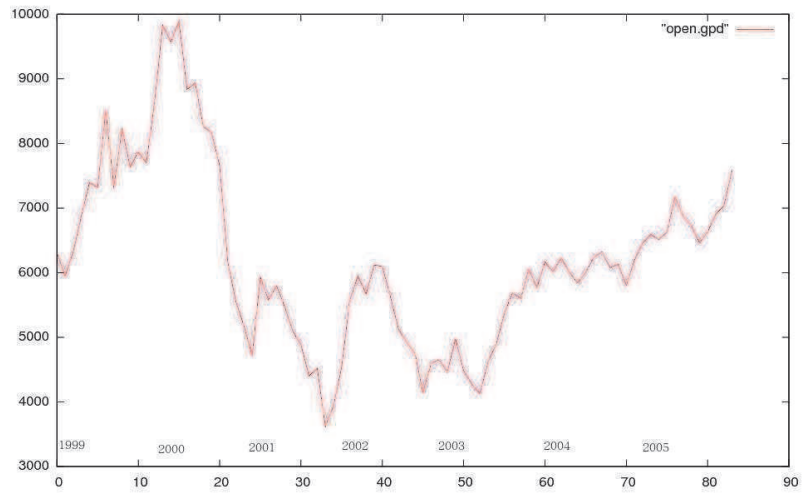


Figure IV.21: Taiwan stock market weighted Index during the period between 1999 and 2006. The index began its drop since 2000, and returned to raising after 2003. Although the lowest point is in 2002, there is another serious drop between 2002 and 2003.

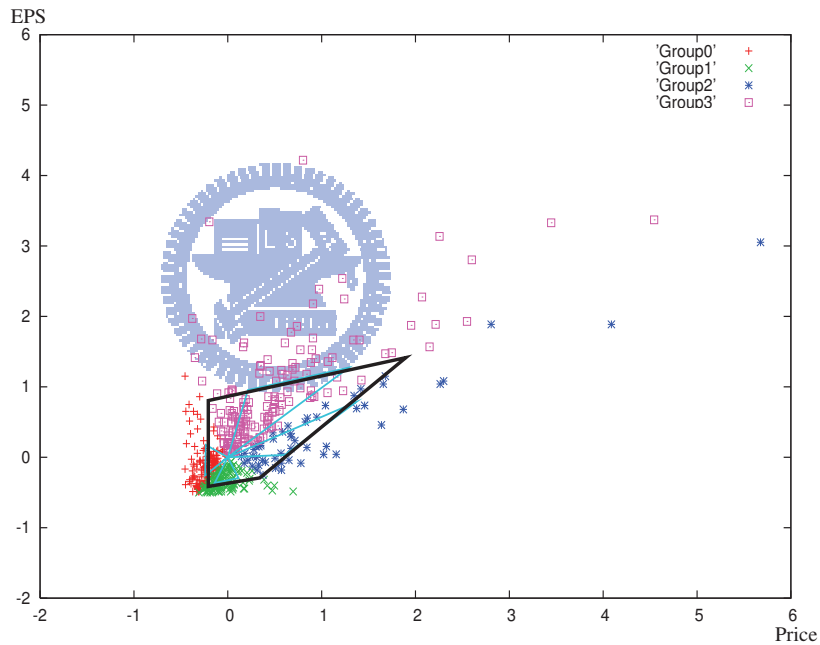


Figure IV.22: An example of a resulted Polygon descriptor for EPS-price data distribution in February 2003. The Polygon descriptor segments input data points into four clusters. Four normal vectors are determined from the four clusters of data points. According to four normal vectors, a contour of the Polygon descriptor is plotted in thick lines.

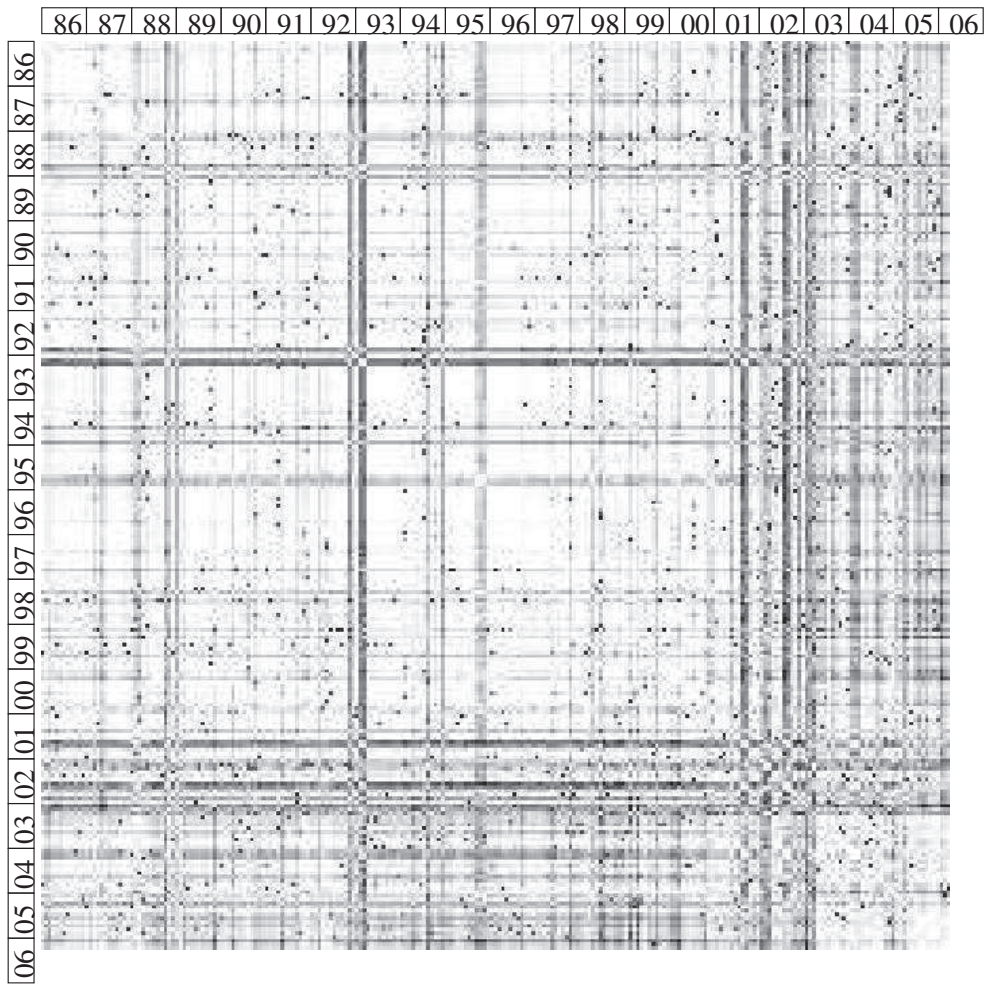


Figure IV.23: The similarities between every two months of TW stock market data between 1986 and 2006. The grey intensity at i -th row and j -th column represents the similarity degree between the i -th month and j -th month since January 1986. Larger grey intensity means higher similarity between data sets.

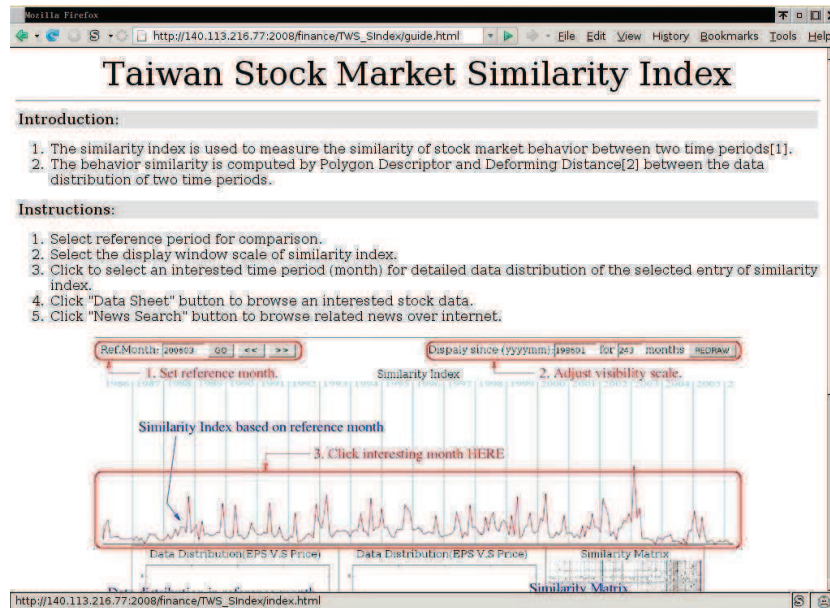


Figure IV.24: The home page of the proposed prototype system. The contents of the home page are the system introduction and user instructions.

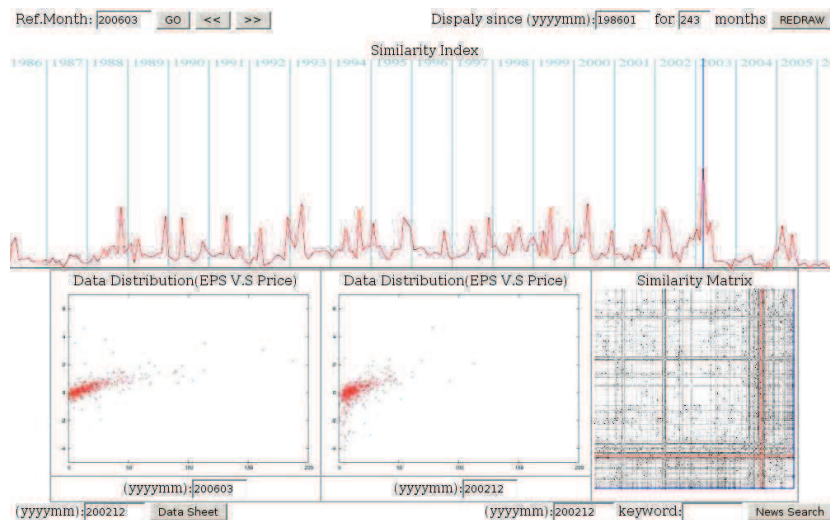


Figure IV.25: The main user interface for the web prototype system. The user interface shows the similarities between reference month and the rest months. By clicking on the region for display of similarities, the data distribution of corresponding month are shown. Then the user can browse the data items by click the button at the left-bottom or browse the related news by click the button at the right-bottom.

ID	Date	Close	Quantity	EPS	Net Assets per share
1101	200109	5.44	86	0.53	17.97
1102	200109	6.24	26	0.31	16.08
1103	200109	6.1	5	0.22	13.93
1104	200109	3.96	37	0.35	18.21
1107	200109	3.17	21	0.2	8.75
1108	200109	2.4	3	0.37	9.99
1109	200109	7.66	1	0.14	12.3
1110	200109	3.51	1	0.14	13
1201	200109	5.8	14	1.67	7.54
1203	200109	3.15	0	0.93	10.81
1204	200109	3.18	0	0.86	8.23
1207	200109	9.91	3	0.58	9.55
1210	200109	5.52	14	2.49	12.02
1212	200109	2.5	2	1.65	5.25
1213	200109	8.56	0	1.34	14.59
1215	200109	4.64	9	4.09	11.39
1216	200109	8.98	80	0.78	12.04
1217	200109	5.3	4	0.86	9.59
1218	200109	3.21	4	0.99	9.89
1219	200109	3.96	0	1.26	11.16
1220	200109	5	3	0.92	13.18

Figure IV.26: The user interface for browsing data items at a specific time period. The time period of interests is assigned automatically. User can tick on the radio boxes to select their desired data items.

News Archives - [News Articles](#) - [Timeline](#) Results 1 - 10 of about 697 for **Taiwan Stock Market**. (0.07 seconds)

TAIWAN STOCK EXCHANGE TO REINFORCE COOPERATION WITH TOKYO COUNTERPART.
Free with registration - Asia Africa Intelligence Wire - AccessMyLibrary.com - Dec 14, 2002
Taipei, Dec. 14 (CNA) The **Taiwan Stock Exchange Corp.** will reinforce its cooperation with the Tokyo **Stock Exchange Inc.**, and the two companies will ...
[Related web pages](#)

TAIWAN STOCK EXCHANGE TO SIGN MEMORANDUMS WITH FOREIGN COUNTERPARTS.
Free with registration - Asia Africa Intelligence Wire - AccessMyLibrary.com - Dec 31, 2002
Taipei, Dec. 31 (CNA) **Taiwan Stock Exchange (TSE)** Chairman Chen Chung said Tuesday that the TSE will sign a memorandum of understanding with its foreign ...
[All 8 related](#) - [Related web pages](#)

TAIWAN STOCK MARKET CLOSES LOWER.
Free with registration - Asia Africa Intelligence Wire - AccessMyLibrary.com - Dec 30, 2002
Taipei, Dec. 30 (CNA) The **Taiwan Stock Exchange (Taie)** closed lower Monday on Wall Street losses amid mounting concerns over US-Iraq tensions and the ...
[PRICES DOWN ON TAIPEI FUTURES MARKET.](#) - Asia Africa Intelligence... - AccessMyLibrary.com (Free with registration)
[STOCKS - TAIWAN SHARES CLOSE LOWER - DEC 30, 2002.](#) - AsiaPulse News - AccessMyLibrary.com (Free with registration)
[America's Intelligence...](#) - Xinhua News Agency - [All 5 related](#) - [Related web pages](#)

Attek to be listed on Taiwan Stock Exchange on Dec. 24.
Free with registration - Asia Africa Intelligence Wire - AccessMyLibrary.com - Dec 17, 2002
Taipei, Dec. 17, 2002 (CENS)-At the price of NT\$76 per share, **Attek Corp.**, a major digital camera maker in **Taiwan**, is scheduled to get listed on **Taiwan** ...
[Related web pages](#)

Japanese shares battered in another turbulent year
Taipei Times - Dec 30, 2002
[Benchmarks in South Korea and Taiwan are about to complete their second losing year in three](#)

Figure IV.27: The user interface to query related news at specific time period. The request is automatically redirected to Google News Archive Search.

Chapter V

Virtual Geometry for Similarity based Clustering

Polygon descriptor can be used to improve the performance of clustering methods, too. Clustering algorithm segments feature space into partitions for every target class. That is, for each class, decision boundaries between clusters divide feature space into isolated regions. For example, a two-means cluster uses a perpendicular hyperplane of line segments between two cluster centers to partition the feature space into two parts. For each class, its related decision boundaries (hyperplane) wrap the sample data points by a polygonal region. For similarity based clustering, such as K-medoid, Polygon descriptor (see Chapter III) can be used to perform virtual geometry computations for the imagine feature points which is implied by similarity value of object pairs. In this chapter, K-medoid method is introduce and Polygon descriptor is used to improve K-medoid algorithm by adapting variance information into data clusters.

K-medoid clustering algorithm is introduce in Section 1 first. Then the variance enhanced K-medoid is proposed in Section 2. In Section 3, a real world application, which groups photo of a web gallery to efficiently render an index page, is demonstrated.

1 Related Works: K-medoid

Given a data point set P , Medoid is the point p_m of the smallest sum of distance from p_m to all the other points p_i in P . Medoid p_m of P , is mathematically defined as follows:

$$p_m = \operatorname{argmin}_{p_i \in P} \sum_{p_j \in P} \mathcal{N}(p_i, p_j),$$

where $\mathcal{N}(p_i, p_j)$ is the norm between two points p_i and p_j .

For one dimensional data, Medoid of P is actually the median. Given a point x , $Q(x)$, the sum of distance from x to all the other points $p_i \in P$, is formulated as follows.

$$Q(x) = \sum_{p_i \in P} \mathcal{N}(p_i, x).$$

Let m_d be the Medoid, and p_i be another data point. According to the definition of Medoid, $Q(m_d) - Q(p_i)$ has to be smaller or equal to zero.

Figure V.1 illustrates the relation between median and Medoid. Let m be the median, and m_d be the Medoid of P . For each point p_i in P , we can observe the followings.

- When p_i is located between median m and Medoid m_d , $\mathcal{N}(p_i, m_d) = \mathcal{N}(m, m_d) - \mathcal{N}(m, p_i)$;
- When median m is located between p_i and Medoid m_d , $\mathcal{N}(p_i, m_d) = \mathcal{N}(p_i, m) + \mathcal{N}(m, m_d)$;
- When Medoid x is located between median m and p_i , $\mathcal{N}(m_d, p_i) = \mathcal{N}(m, p_i) - \mathcal{N}(m, m_d)$.

$Q(m_d) - Q(m)$ can be derived and rewritten as follows,

$$Q(m_d) - Q(m)$$

$$\begin{aligned}
&= \sum_{p \in (-\infty, m]} \mathcal{N}(m, m_d) - \sum_{p \in [m_d, \infty)} \mathcal{N}(m, m_d) + \sum_{p \in (m, m_d)} \mathcal{N}(p, m_d) - \mathcal{N}(m, p) \\
&= \sum_{p \in (-\infty, m]} \mathcal{N}(m, m_d) - \sum_{p \in [m_d, \infty)} \mathcal{N}(m, m_d) + \sum_{p \in (m, m_d)} (\mathcal{N}(p, m_d) - \mathcal{N}(m, m_d) + \mathcal{N}(p, m_d)) \\
&= \sum_{p \in (-\infty, m]} \mathcal{N}(m, m_d) - \sum_{p \in (m, \infty)} \mathcal{N}(m, m_d) + 2 \sum_{p \in (m, m_d)} \mathcal{N}(p, m_d) \\
&= \mathcal{N}(m, m_d) + 2 \sum_{p \in (m, m_d)} \mathcal{N}(p, m_d)
\end{aligned}$$

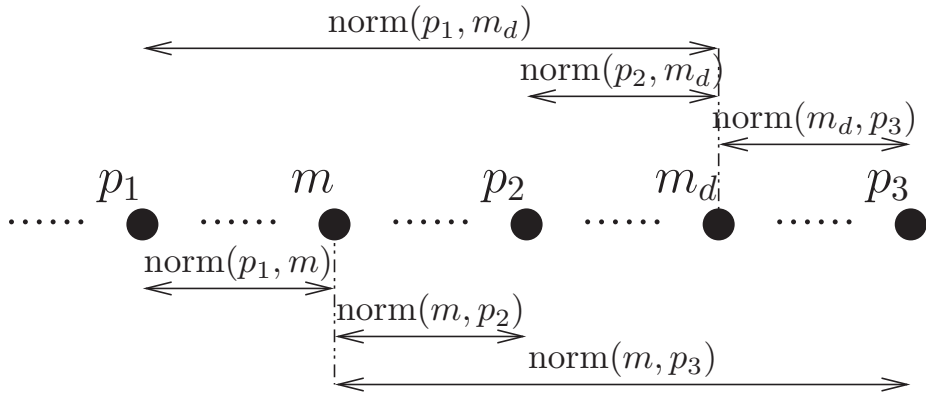


Figure V.1: The relation between Medoid and median for one dimensional data points. Let m be median and m_d be Medoid. Since Medoid m_d is the minimal of $Q(m_d)$, Medoid m_d is equal to median m .

Since $Q(m_d) - Q(m) \geq 0$, and $Q(m_d)$ is the minima (the definition of Medoid), the Medoid m_d must be equal to median m . The same results hold when m_d is smaller or equal to m . Therefore, for one dimensional data, Medoid is equal to median.

Intuitively, Medoid has higher chance to be located in high density region than mean. Therefore, Medoid may be a better representation of a data set.

Similar to the popular K-means algorithm, K-Medoid[12] can be used to cluster data points into K groups. Unlike K-means algorithm, K-medoid does not depend on the coordinates or values of each data points in feature space, the only required data for Medoid estimation is the norm between every pair of elements.

Given K cluster centers $M = \{m_1, \dots, m_j, \dots, m_K\}$, and data points in $P = \{p_1, \dots, p_i, \dots, p_N\}$ are partitioned into K clusters. For data point p_i , a cluster center m_w in M with the minimal norm between m_j and p_i , is determined according to the following equation,

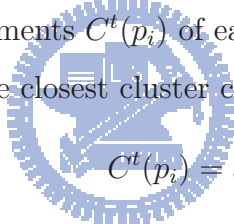
$$m_w = \underset{m_j \in M}{\operatorname{argmin}} \mathcal{N}(m_j, p_i),$$

where $\mathcal{N}(m_j, p_i)$ is the norm between m_j and p_i . For each data point p_i , by assigning it to the cluster with the minimal norm between p_i and the Medoid of each cluster, a data set P can be partitioned into K clusters.

1.1 The Learning Algorithm of K-medoid Clustering

The K-medoid cluster learning algorithm[12] is briefly stated as follows. Let $\mathcal{N}(p_i, p_j)$ be the norm between p_i and p_j in data set P , and K is the number of Medoid.

1. Given a current set of cluster centers $M^t = \{m_1^t, \dots, m_k^t\}$, the current cluster assignments $C^t(p_i)$ of each data points $p_i \in P$ are evaluated by computing the closest cluster center for each data point p_i as follows:



$$C^t(p_i) = \underset{m_j^t \in M^t}{\operatorname{argmin}} \mathcal{N}(m_j^t, p_i),$$

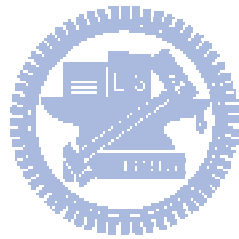
where $\mathcal{N}(m_j, p_i)$ is the norm between m_j and p_i , and $C^t(p_i)$ is the cluster assignment for p_i .

2. Given a set of current cluster assignments C^t , the new cluster centers $M^{t+1} = \{m_1^{t+1}, \dots, m_j^{t+1}, \dots, m_k^{t+1}\}$ are estimated by computing the Medoid among the data points in the same assignment as follows:

$$m_j^{t+1} = \underset{C^t(p_x)=m_j^t}{\operatorname{argmin}} \sum_{C^t(p_y)=m_j^t} \mathcal{N}(p_x, p_y),$$

where $C^t(p_i)$ is the cluster assignment for p_i .

3. Iterate steps 1 and 2 until the assignments approach to unchange.



1.2 Self-growing K-medoid

Generally speaking, asking user to provide the number of required clusters is not practical since the number of clusters is usually the natural of data distribution and is often unknown to a user. That is, a user may have to examine to the data distribution to determine how many clusters are required to partition the data distribution.

In order to perform data clustering without predetermined cluster number, the Self-growing K-medoid clustering algorithm is proposed. By giving the maximal acceptable dis-similarity between a data point and its cluster center, the proposed method iteratively increases the number of clusters until the dis-similarity between any data point and the associated cluster center is below a given threshold.

The idea of the proposed method can be illustrated by the example shown in Figure V.2. The y-axis shows the value of weight for each points, the x-axis is the location of data points. Based on K-medoid clustering algorithm, the point p_3 is first selected to be the first cluster center c_1 . The weights of each data point are updated according to their norms with respect to the selected cluster center. Specifically, the weights of data points that are close to the selected cluster center will be decreased more than the data points that are far away from the selected cluster center. Then another data point locates in the middle of data points with larger weights is selected as the second cluster center c_2 . After clustering all data points with the two selected cluster centers, the weights of data points are updated. The clustering processes are repeated to decrease the weights of data points until all weights are below to a given threshold.

To formulate this idea, a weighted version of Medoid estimation is defined

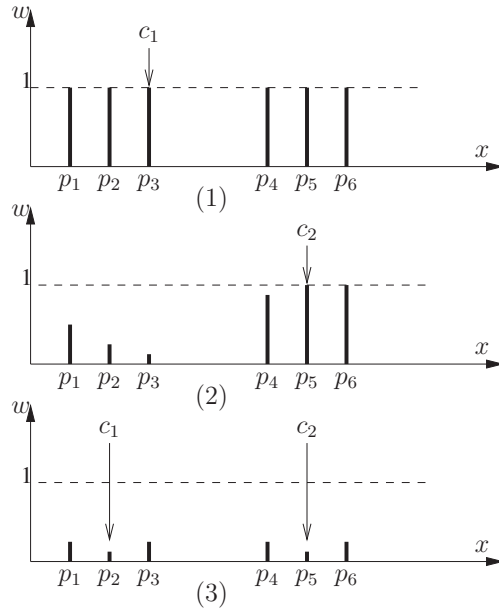


Figure V.2: A 1-D data distribution example is used to illustrate the idea of Self-Growing K-medoid Clustering algorithm. In each diagram, the heights (w) of a sample points represents its weight values. (1) At first, the middle point is selected as the first cluster center c_1 ; (2) By decreasing the weights of data points close to the first cluster center, a new cluster center p_5 in the middle of data points with large weight values is selected to be the new cluster center c_2 ; (3) After refining the location cluster centers by standard K-medoid clustering algorithm, the weights (the dis-similarity between data point and the closest cluster center) of all data points are below to a given threshold.

as follows,

$$p_n = \operatorname{argmin}_{p_j \in P} \left(\sum_{p_i \in P} w_j w_i \mathcal{N}(p_j, p_i) \right).$$

The weighted version of Medoid estimation can be used to organize a new cluster center for data points with high weight values. Based on the weighted version of Medoid estimation and the standard K-medoid algorithm, the self growing K-medoid method is depicted in Figure V.3.

As the flow chart shown in Figure V.3, a weight value w_i of each data point p_i is initialized to be 1. According to weight values, the first cluster

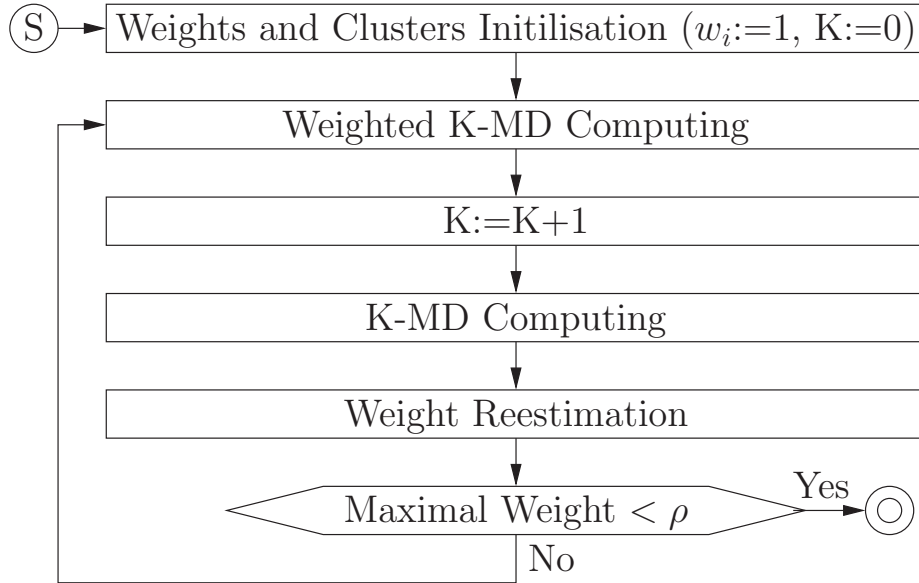


Figure V.3: The flow chart of Self Growing K-medoid method. The number of clusters is growing until the minimal dis-similarity between data objects and cluster centers (Medoid) is small enough. Since the weight of each data point can represent the minimal dis-similarity between data objects and cluster center, Self Growing K-medoid method grows clusters until all the weights are smaller than a threshold ρ . During the iterative cluster processing, the minimal dis-similarity of data objects are stored and used to weight the data objects for generating the new cluster center. By interactively generating new cluster centers to increase the number cluster centers, the K-medoid model automatically adjust itself to have the minimal dis-similarity between data objects and cluster centers.

center is estimated by using the weighted version of Medoid estimation. Then the standard K-medoid clustering algorithm is applied to refine the location of clusters. According to the location of cluster centers, the weights of each data points are recomputed as follows,

$$w_i = 1 - \left(1 + \min_{p_m \in M} \mathcal{N}(p_i, p_m)\right)^{-1}$$

Then, another new cluster center is estimated using the updated weights. By repeatedly generating new clusters by using the weighted version of Medoid estimation and adjusting cluster centers by using the standard K-medoid

algorithm until all the weights are small enough and no new clusters can be generated. Then, the number of clusters converges.



2 Variance Enhanced K-medoid

2.1 Variance for K-medoids

Based on the mathematical representation of two clusters, a decision boundary can be determined to separate data points into two parts. As shown in Figure V.4 (a), the decision boundary between two K-medoid clusters is the perpendicular bisectors of a line from one cluster center to another. Since the data variance is not considered, these decision boundaries do not separate data clusters properly. To achieve a better separation of the data points, variance should be considered in determining the decision boundaries. A variance-enhanced K-medoid is illustrated in Figure V.4 (b). By aggregating the norm between data points and its cluster center (medoid), the variance of each cluster is estimated. Again, since the variance orientation of data distribution is not considered, the decision boundaries still does not partition the feature space well enough. For instance, without considering variance orientation, a cluster with widely data distributed in one direction and with vary little data distributed in a perpendicular direction, may result a large overall variance along the perpendicular direction. Figure V.4 (c) shows the decision boundaries in corresponding to data cluster with multiple variances being considered along several orientations. Based on the location of data points in feature space, a K-means model can be improved by including the covariance matrix for each cluster, e.g., Mixture Gaussian Model. Since the location information of data points is not used in K-medoid model, improving the K-medoid model using the covariance matrix method seems impracticable. How to include the oriented variance in K-medoid method, and how to estimate the variances in various orientations become essential issues for variance enhanced K-medoid algorithm. The detailed description of this method

will be illustrated in Section 2.

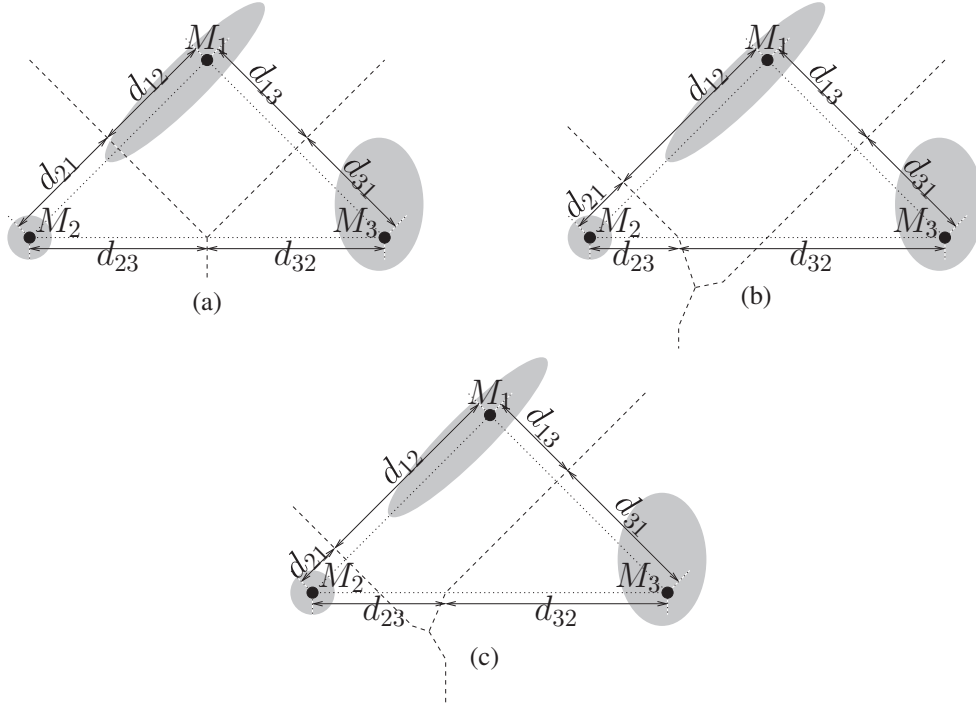


Figure V.4: The comparison among three kinds of decision boundary for clustering model. Shaded area are the distributed regions of data points. (a) The decision boundary is the perpendicular bisector of the line between two cluster centers. The variance of data distribution is not used in creating data clusters. (b) A single variance for every direction is used for each cluster. For two cluster, the ratio between the two variances of clusters is equal to the ratio between the distances from cluster centers to decision boundary. (c) For each two clusters, the location of decision boundary is decided according to the data variances along the line between two cluster centers.

2.2 The Learning Algorithm of Variance Enhanced K-medoid

The learning process is based on some concepts, which will be introduced first. Then the learning process is concluded in Section 2.2.

Center-to-Center Variance

Based on the discussion in Section 2.1, multiple variances along several orientations are required for K-medoid based clustering method. However, since the locations of data point in feature space are not available, estimating variance orientation by using covariance matrix seems impractical. Therefore, multiple variances along the lines between cluster centers are proposed instead. Let's see that, the variance along a line between cluster centers should be measured according to the data points located along the line. As the example shown in Figure V.5, four cluster centers M_1, M_2, M_3, M_4 are shown, and b_i 's are the decision boundaries between the lines from M_1 to M_i for $i = 2, 3, 4$. For instance, the variance along $\overline{M_1M_3}$ associated with cluster center M_1 is estimated by the data points located in the region of gray-scaled color. In general, decision boundaries segment the feature space into polygonal regions for each cluster in a data set. And, the polygonal region for a cluster can be further divided into several smaller pyramid-shape segments, whose tops are the cluster centers and bottom hyperplane are along the decision boundaries. The variance along the line $\overline{M_1M_3}$ is measured by using data points located in the pyramid-shape segment, whose bottom edge is along the decision boundary b_3 between centers M_1 and M_3 .

To select data points for the computation of variance along lines between cluster centers, Polygon descriptor proposed in Chapter III is suggested to cluster data points into side clusters. The data points are clustered into

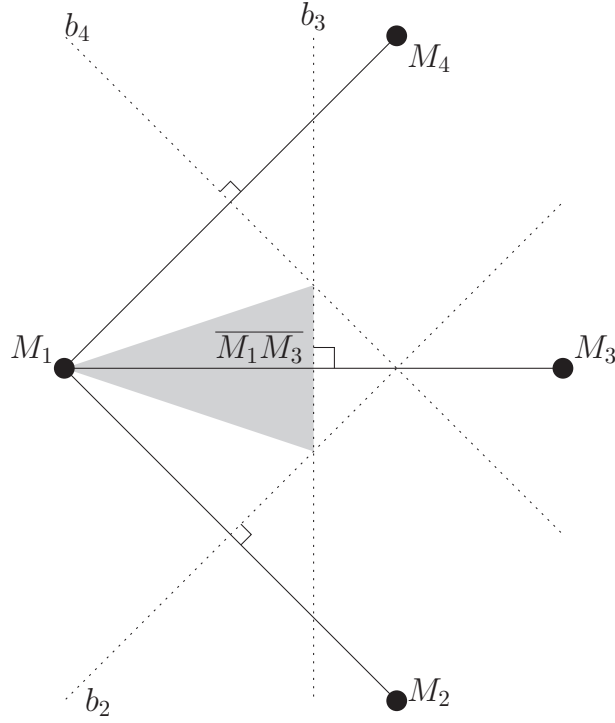


Figure V.5: Giving four clusters with centers at M_1 , M_2 , M_3 , and M_4 . For the center M_1 , there are three associated decision boundaries b_2 , b_3 , and b_4 . The variance between M_1 and M_3 are computed by using the data points inside the pyramid-shape segment whose top is located at the cluster center M_1 and the bottom edge is along with decision boundary b_3 .

side clusters by using Polygon descriptor at first. The data points in each side-cluster are projected to the line between cluster centers by the method proposed in Section 2.2. Then, Section 2.2 proposed methods to adjust the decision boundaries between clusters according to the variance of data distribution along lines between cluster centers.

As shown as Figure V.4, decision boundaries segment feature space into several polygonal regions (data clusters). The proposed Polygon descriptor can be used to model a polygonal data distribution.

Let $\{\vec{a}_1, \dots, \vec{a}_j, \dots, \vec{a}_M\}$ be the M normal vectors of a polygon descriptor. M normal vectors of a polygon descriptor can be used to further segment a

data cluster into M sub-clusters, named side-clusters. Given a set of data points $p_i \in P$, a sub-cluster assignment $C(p_i)$ for a data points p_i can be formulated as follows,

$$C(p_i) := \operatorname{argmax}_{j=1}^M \frac{\vec{p}_i \cdot \vec{a}_j}{\vec{a}_j \cdot \vec{a}_j}$$

where $C(p_i)$ is cluster assignment for p_i , and \vec{p}_i is the vector from reference center to p_i . According to the cluster assignments, data points are partitioned into M side-clusters.

By segmenting the data points of a cluster into side clusters corresponding to the lines between cluster centers and projecting the data points in a side cluster to the corresponding line between cluster center, a projected 1-D data distribution can be created. The 1-D data distribution can be used to measure the variance along the line between cluster centers or adjust the location of decision boundary between clusters. However, since the coordinates of data points for K-medoid model is absent, a special method is proposed in Section 2.2 to estimate the projection length without the coordinate of data points.

Projections on the Center-to-Center Link

Computing variance along normal vector by using projection ratio requires coordinate of data points in feature space. In order to computing the projection ratio on normal vectors without the coordinates of data points in feature space, a new projection method is proposed.

As shown in Figure V.6, given a point p , and two clusters with center points at m_1 and m_2 , $\|\overrightarrow{m_1 p_i}\|$, $\|\overrightarrow{m_2 p_i}\|$, and $\|\overrightarrow{m_1 m_2}\|$ are the norms from p to m_1 , p to m_2 , and m_1 to m_2 , respectively. Let d be the projection of p on $\overrightarrow{m_1 m_2}$. $\|\overrightarrow{m_1 d}\|$ is the projection length of $\overrightarrow{m_1 p_i}$ on $\overrightarrow{m_1 m_2}$. As Figure V.6

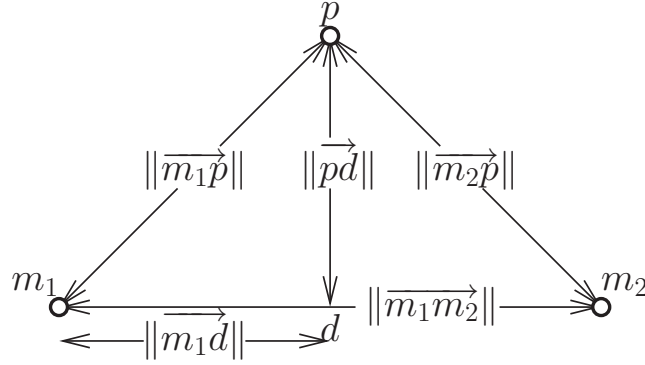


Figure V.6: m_1 and m_2 are two cluster centers. p is a data point. d is the projection point on $\overline{m_1m_2}$.

shown, the following relation holds.

$$\begin{aligned}\|\overrightarrow{m_1p_i}\|^2 &= \|\overrightarrow{m_1d}\|^2 + \|\overrightarrow{p_id}\|^2 \\ \|\overrightarrow{m_2p_i}\|^2 &= (\|\overrightarrow{m_1m_2}\| - \|\overrightarrow{m_1d}\|)^2 + \|\overrightarrow{p_id}\|^2\end{aligned}$$

Then the length of $\|\overrightarrow{m_1d}\|$ can be derived as follows.

$$\begin{aligned}(\|\overrightarrow{m_1m_2}\| - \|\overrightarrow{m_1d}\|)^2 &= \|\overrightarrow{m_2p_i}\|^2 - \|\overrightarrow{m_1p_i}\|^2 + \|\overrightarrow{m_1d}\|^2 \\ \|\overrightarrow{m_1m_2}\|^2 - 2\|\overrightarrow{m_1m_2}\|\|\overrightarrow{m_1d}\| &= \|\overrightarrow{m_2p_i}\|^2 - \|\overrightarrow{m_1p_i}\|^2 \\ \|\overrightarrow{m_1d}\| &= \frac{\|\overrightarrow{m_1m_2}\|^2 - \|\overrightarrow{m_2p_i}\|^2 + \|\overrightarrow{m_1p_i}\|^2}{2\|\overrightarrow{m_1m_2}\|}\end{aligned}$$

By projecting data points along the line between two cluster centers, the 1-D data distribution along the line becomes available. Based on the 1-D data distribution, the decision boundary can therefore be adjusted to maximize the margin the margin between two estimated clusters.

Decision Boundaries Adjustment

By using the projection method proposed in Section 2.2, data points in cluster can be segmented into several side-clusters corresponding to the lines between each pair of cluster centers. The projection of data points in a side-cluster on

the corresponding line between cluster centers forms a 1-D data distribution. Based on the 1-D data distribution, the variance of data points along the line between cluster centers can be measured. Since the purpose of measuring variance is to adjust the location of decision boundary, the estimation or the measuring of variance can be avoided by directly adjusting the location of decision boundary according to projected 1-D data distribution.

As shown in Figure V.7, given two cluster centers m_1 and m_2 . By projecting data points p_i (shown in black square) on the line $\overline{m_1 m_2}$, a series of 1-D location values (shown in black triangle) are measured. The following iterative procedure is proposed to find a decision boundary, such that the distance between the means of two partitioned clusters can be maximized. These 1-D values are separated into two groups g_1 and g_2 by a decision boundary b_{12} . The decision boundary b_{12} is determined to maximize the distance between μ_1 and μ_2 . μ_1 and μ_2 are the means of g_1 and g_2 respectively.

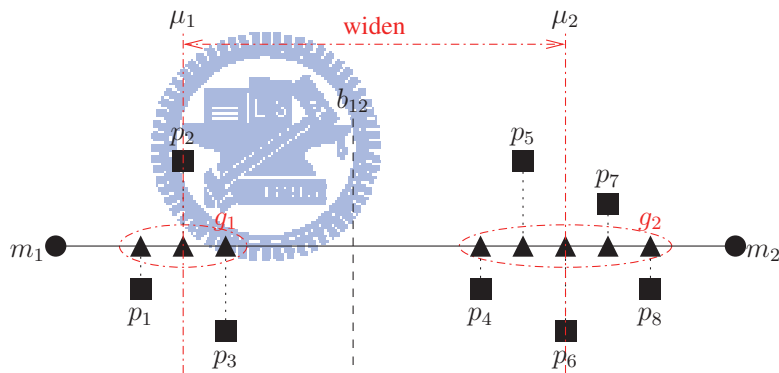


Figure V.7: Data points p_i are projected to the line between cluster centers m_1 and m_2 . The decision boundary is located at the place which partition the projected data points (triangle points) into two groups with farthest mean location μ_1 and μ_2 .

Let α be the target location of decision boundary and the projected ratio

$h(p_i)$ is defined as

$$h(p_i) = \frac{m_1 \vec{p}_i \cdot m_1 \vec{m}_2}{m_1 \vec{m}_2 \cdot m_1 \vec{m}_2}.$$

Actually, $h(p_i)$ is the projection length of vector $m_1 \vec{p}_i$ on $\overline{m_1 m_2}$ dividing by the distance between two cluster centers. Then, α can be evaluated by the following formula:

$$\alpha = \operatorname{argmax}_{\alpha} \left(\frac{\sum_{h(p_i) > \alpha} h(p_i)}{\sum_{h(p_i) > \alpha} 1} - \frac{\sum_{h(p_i) < \alpha} h(p_i)}{\sum_{h(p_i) < \alpha} 1} \right),$$

where $\frac{\sum_{h(p_i) > \alpha} h(p_i)}{\sum_{h(p_i) > \alpha} 1}$ is the average project ratio that is larger than α , and $\frac{\sum_{h(p_i) < \alpha} h(p_i)}{\sum_{h(p_i) < \alpha} 1}$ is the average project ratio that is smaller than α .

The detail process to estimate the decision boundary between two clusters are illustrated as follow. Data points are clustered by using the K-medoid method proposed in Section 1.2. For every two clusters C_i and C_j , for $j > i$, data points in C_i and C_j are projected to $\overline{m_i m_j}$, where m_i and m_j are Medoid of C_i and C_j respectively. Then the projection length is partitioned into 10 intervals. By counting the number of points locating in one interval, a histogram is built. Based on the histogram, the decision boundary is located so that the distance between two means of data sets, which are separated by the decision boundary, are maximized. All data points are then redistributed using the adjusted decision boundaries. The process repeats until the location of decision boundaries converged, such that the distance between μ_1 and μ_2 is maximized.

Algorithm

The flow chart of variance enhanced K-medoid is shown in Figure V.8. At first, one Medoid is estimated. Then the number of Medoid is gradually increasing until the distance from each data point to the closest cluster center is smaller than a given threshold. For each Medoid, its positions are estimated

using general K-medoid algorithm first. Then the 1-D data distributions can be established by projecting data points in side-clusters to the corresponding line between cluster centers. According to these 1-D data distributions, the location of decision boundaries are adjusted to redistributed data points to each clusters. After the position of Medoid converged, weights w_i of each points p_i is calculated as follows

$$w_i = 1 - \left(1 + \min_{p_m \in M} \text{sim}(p_i, p_m) \right)^{-1},$$

where M is a set of estimated Medoid, and $\text{sim}(p_i, p_m)$ is the projected similarity defined as follows,

$$\text{sim}(p_i, p_m) = \underset{p_j \in M, p_j \neq p_m}{\text{argmax}} \frac{\vec{p}_i \cdot (\vec{p}_j - \vec{p}_m)}{\alpha(\vec{p}_j - \vec{p}_m) \cdot \alpha(\vec{p}_j - \vec{p}_m)}$$

where $\alpha(\vec{p}_j - \vec{p}_m)$ is the vector from p_m to the decision boundary between the clusters associated to p_m and p_j . Based on these weights, a new Medoid is created to include high weighted data points. By repeating these processes, the number of Medoid gradually increasing until the maximal weight is lower than a given threshold.

By using the proposed Variance Enhanced K-medoid, the location of decision boundaries between each pair of cluster centers are related to the data distribution along a line between cluster centers now. A real-world application, photo album preview with variable sized thumbnail, are proposed in Section 3 to illustrate the performance of the proposed variance enhanced K-medoid method.

3 Real World Application

The proposed variance enhanced K-medoid can be used to create a variable sized thumbnail index page for web gallery system. By using a variable-sized

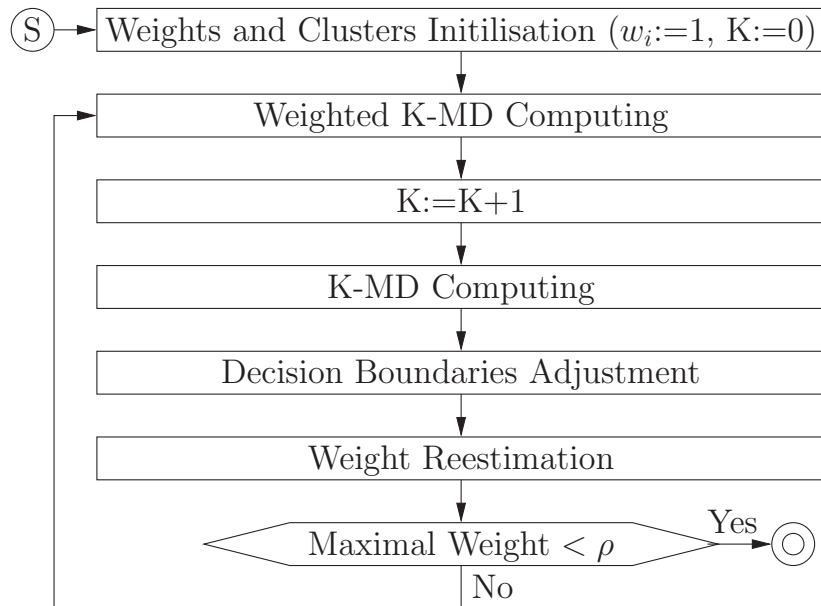


Figure V.8: The flow chart of variance enhanced K-medoid clustering. At first, general K-medoid algorithm is applied. According to the resulted clusters of general K-medoid algorithm, the cluster-to-cluster variance are estimated. Then the weights for each data points are updated and new cluster is estimated based on the weighted data points. These steps are repeated until all weights are smaller than a given threshold ρ .

thumbnail index page, the thumbnail index page can display more thumbnail in a single page, and help user browser a set of photo in groups based on image similarity.

3.1 Variable-sized Thumbnail of Web Gallery Service

Variance enhanced K-medoid, which is introduced in Chapter V, can be used to intelligently create image thumbnail, where "intelligently" means that the thumbnail(preview) images are created in different sizes automatically. Generally, image gallery creates thumbnail to provide a quick preview for each photo. However, even though the thumbnail are down-sized from the original photos, the display area is often still too small to contain all thumbnail. To

contain all thumbnail of one size in a page, its very difficult to select a proper size for the thumbnail. Sometimes, it may be too small to clearly represent the original image, or it may be too large to put all thumbnail in a page. Therefore, variable sized thumbnail is needed to fit all thumbnail into a page without losing too much details.

Since plenty of images in an image gallery may be similar to each other, these images can be clustered into groups according to their similarity. For each group, a most representative image can be depicted by a normal sized thumbnail, and the rest of similar images can be shown by smaller thumbnail.

The detail description of the proposed system is depicted in the following sections. Color information of an image is used to create four bitmaps to show the distribution of dominant color components. Based on these bitmaps, the similarity between two images are measured. Then the proposed variance enhanced K-medoid is applied to cluster all images into groups.

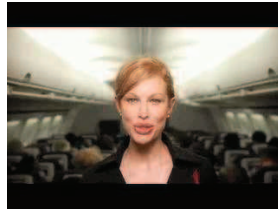
3.2 Color Clustering

Giving an image I , K dominant color components are selected to establish K bitmaps which show the coverage for data distribution of dominant color components. K-means algorithm is used to cluster the color value of each pixel $p_{x,y} \in I$. Let $K=4$, then four dominant colors m_i , for $i = 1, \dots, 4$, are estimated by K-means algorithm. Then, four bitmaps $I(x, y)_i$ are created as follows,

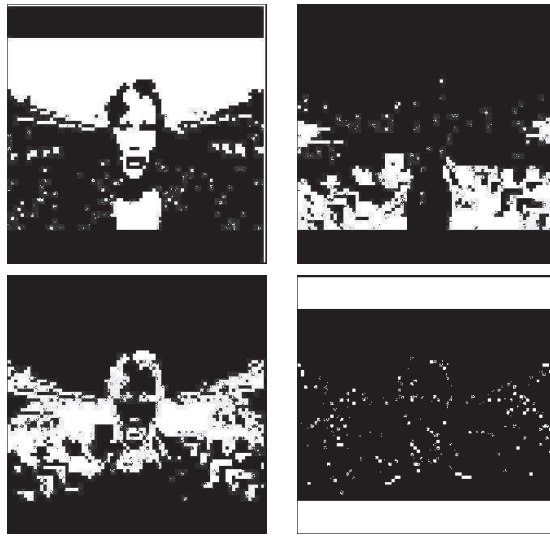
$$I(x, y)_i = \begin{cases} 1 & , \text{ if } \operatorname{argmin}_{m_i} \operatorname{norm}(m_i, c_i) = m_i \\ 0 & , \text{ if } \operatorname{argmin}_{m_i} \operatorname{norm}(m_i, c_i) \neq m_i \end{cases} \text{ for } i=1, \dots, 4;$$

where $I(x, y)_i$ is the bitmap created using dominant color m_i , and c_i is the color at (x, y) .

An example is shown in Figure V.9, the size of each image is normalized to 80x80 at first. Then K-means algorithm are used to cluster pixels of



(a)



(b)

Figure V.9: By using K-means algorithm, pixels of original image, for example (a), are clustered into four dominant groups according its color value. Then four bitmaps, for example (b), are established to show the data distribution of pixels in each dominant groups.

original image to four groups according to the color values of pixels. And, four bitmaps are established to show the distribution of data points in each groups. These four bitmaps will be used to measure the similarity between images (see Section 3.3).

3.3 Similarity between Images

Since a Images is separated to several bitmaps which show the distribution of main dominant colors, the similarity measurement between images are based

on the similarity between every pair of bitmaps. Given two normalized bitmaps which show the coverage area for data distribution of specified color components, the similarity $S(A, B)$ between two bitmaps A and B , is defined as follows,

$$S(A, B) = \frac{\|\mathcal{A} \cap \mathcal{B}\|}{\|\mathcal{A} \cup \mathcal{B}\|},$$

where \mathcal{A} and \mathcal{B} are the sets containing the pixels with value 1 in bitmap A and B respectively. As shown as Figure V.10, the similarity is defined according to the ratio between overlapped (δ) and overall ($\alpha + \delta + \beta$) area of \mathcal{A} ($\alpha + \delta$) and \mathcal{B} ($\beta + \delta$).

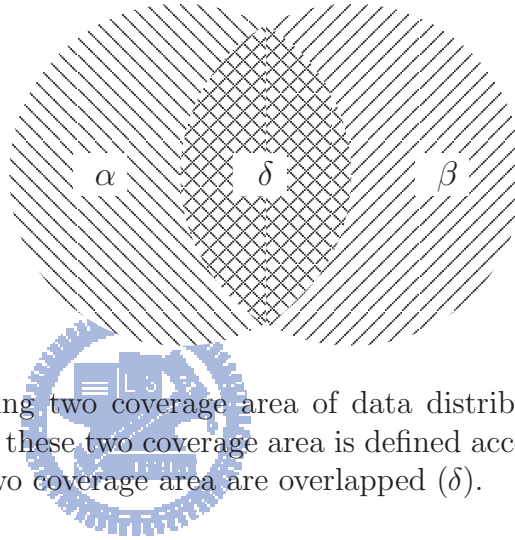


Figure V.10: Given two coverage area of data distribution α and β , the similarity between these two coverage area is defined according to how much portion of these two coverage area are overlapped (δ).

Since four bitmaps are extracted to represent original image, the similarity between two images can be estimated based on the group similarity S_g between two group of bitmaps. Based on the matching algorithm proposed in Section 3, the best match between the elements in bitmap groups can be found.

According to the best match, the group similarity S_g is defined as follows,

$$S_g = \sum_i \frac{\|\mathcal{A}_i\| + \|\mathcal{B}_i\|}{\sum_j (\|\mathcal{A}_j\| + \|\mathcal{B}_j\|)} \times S(A_i, B_i),$$

where \mathcal{A}_i and \mathcal{B}_i are the sets containing the pixels with value 1 in bitmap A_i and B_i respectively. That is, the group similarity S_g is the weighted sum of similarities between each pair of bitmaps, according to the total number of pixels in each pair of bitmaps.

For each image, four bitmaps are extracted to represent the characteristics of image. To measure the similarity between two images, the method introduced in Section 3.1 is applied to find a *minimal-cost* one-to-one onto match between elements of two bitmap groups.

3.4 Images Clustering

Given an image set $\mathcal{I} = \{I_1, \dots, I_N\}$, a matrix \mathcal{M} is constructed by measuring the similarity between any two images using the method proposed in Section 3.2, and 3.3. The image similarity matrix \mathcal{M} for image comparison is defined as follows,

$$\mathcal{M} = \begin{bmatrix} m_{11} & \cdots & m_{1N} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NN} \end{bmatrix}, \text{ where } m_{ij} = S_g(I_i, I_j).$$

Then the proposed *variance enhanced K-medoid* model clusters images by using the pair-wise image similarity values S_g in a image similarity table \mathcal{M} . The image associated with the medoid of a data cluster is used to be the representative image for each image cluster.

3.5 System Demonstration

A web-based prototype system[1] is implemented to demonstrate the proposed method for public trial and testing. As shown in Figure V.11, a user may select a test media set from the selector at the top-right corner. The threshold which represents the maximal allowed difference between data objects and its cluster center, is selected by clicking at a proper position on

the ruler at the top-left corner. Then, the images in the test media set is clustered. And, the resulted clusters are presented in the order of cluster size. For each cluster, the represented image of each cluster is displayed in larger size, and the rest of images in a cluster are shown in smaller sized thumbnail.

Figures V.11, V.12 and V.13 show the testing results using three different thresholds on the same media data set. As we can see, using higher threshold may reduce the number of clusters. In other words, enlarging the similarity threshold reduces the similarity among data elements in certain clusters, so as to receive more less-similar data elements. That is, their maximal allowed variance is increased. However, for data clusters containing quiet similar data elements, enlarge the similarity threshold will only slightly increase the number of less similar data elements, since the maximal allowed variance has little inference on these data clusters. Thus, the noisy data elements are clustered into certain clusters instead of spreading to all clusters.



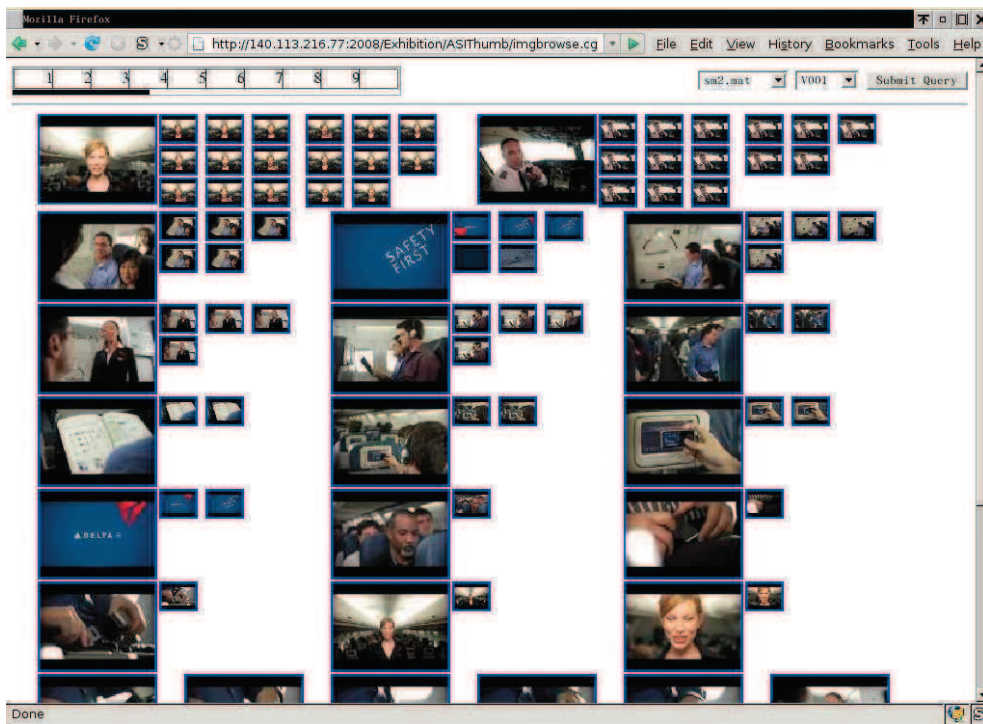


Figure V.11: According to the web-based prototype system, the proposed method successfully cluster similar images together. In this demonstration, the similarity threshold is set as 0.35. That is, the similarity between a data object and the representative object (cluster center) of a cluster should be larger than $0.65 (= 1 - 0.35)$.

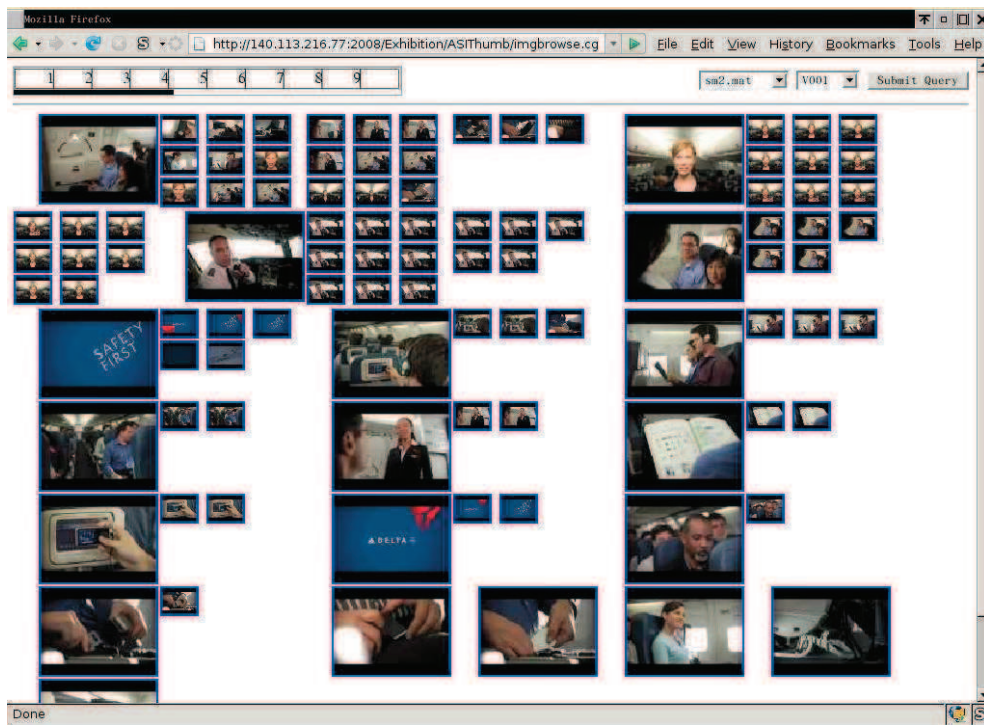


Figure V.12: By increasing the similarity threshold, certain data clusters receive several less-similar data objects. However, the data clusters, which are previously contained quiet similar data elements, still have almost the same similarity among each other.

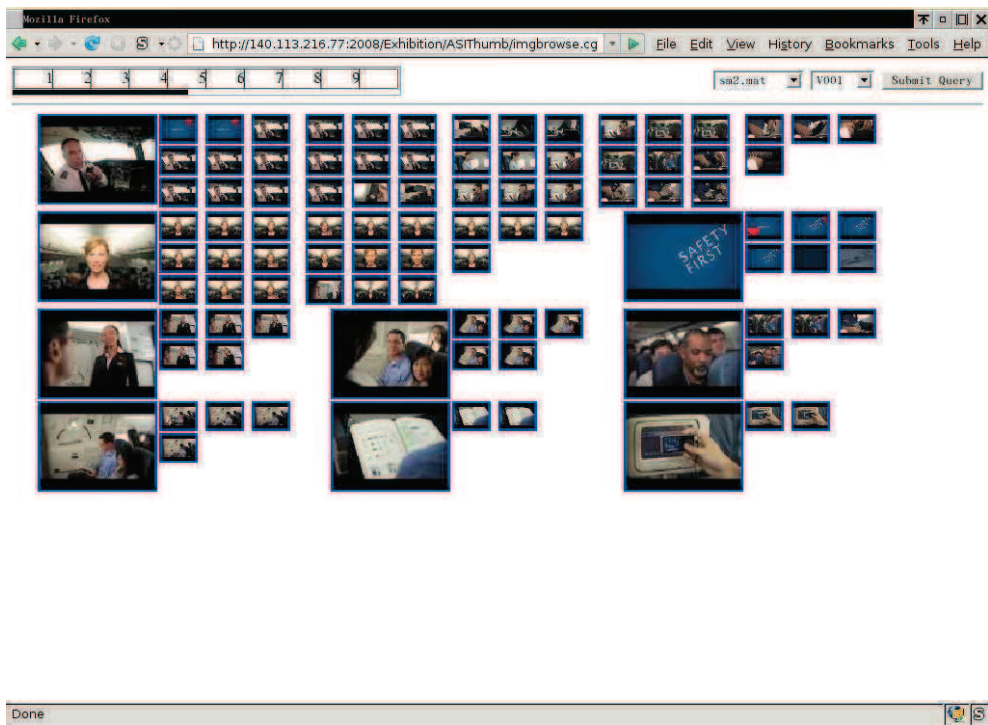


Figure V.13: By increasing the similarity threshold again, data clusters previously contained less-similar objects will grow to accept more less-similar objects. However, these data clusters which previously contained quiet similar objects will still have tightly similar objects.

4 Concluding Remarks

In this chapter, Polygon descriptor is applied to improve K-medoid algorithm by adapting variance information to data clusters.

On the other hand, the improvement of clustering based on Polygon descriptor can improve the power of Polygon descriptors too. K-medoid algorithm groups data objects into clusters according to the similarity between data objects. K-medoid method segments feature space into several convex regions. For data points in each region, polygon descriptors can be measured to represent the shape of data distribution in each region. By merging adjacent polygon descriptors, polygon descriptors can therefore be used to model data distribution in concave shape.



Chapter VI

Polygon-based Region Representation

Polygon descriptor can also be used to represent a polygon-shape regions. A polygonal region selector is proposed and demonstrated in this section. A potential application of using a polygonal model is object tracking. Based on the deformation of estimated polygon region, various movement of an object can be tracked, including translation, rotate, flipping, and so on. However, in this section, only the method of how to describe a set of pixels by a polygon descriptor is introduced. An object tracking application is depending on the criteria of real world applications.

1 Polygon Region Selection

Figure VI.1 shows the flow chart of the proposed polygon-shape region selector. Giving an image and an initial location, the color of pixel at initial location is selected as the reference color. At first, the distance in *CIELAB* color space between each color of pixel and reference color are measured. Then all pixels are sorted according to the color difference. By using the sorted pixel set, a part of pixels with small color difference are selected as

sample points.

Then the overall learning process started from using the standard learning process to estimate a polygon descriptor from sample points. The estimated Polygon descriptor is used to weight the sample points and be down-sized to serve as the initial model of another Polygon descriptor learning process, which using fixed number of normal vectors. Since the sample data points are weighted now, the *discontinuous* data points distributed in outer area are going to be ignored. Then the sample data points are weighted by a new estimated Polygon descriptor and another learning process is applied again by using the down-sized Polygon descriptor as initial model. The weighting, down-sizing, and learning process repeats until the estimated Polygon descriptor converges.

At last, a complete learning process of Polygon descriptor is applied to learn the polygon region of weighted sample data points. Since most outliers are not covered by the initial Polygon descriptor now, most outliers are ignored in the last learning step. The number of normal vectors are also adjusted to fit the shape of data distribution in the last learning step.

Force the other learning steps except the last one, using fix numbered boundaries, can greatly improve the efficiency of learning process. The self-growing estimation of Polygon Descriptor may require more computation power than adjusting fix numbered normal vectors, because it has to repeated splitting and merging normal vectors.

2 Evaluation and Experimental Results

This method is tested by using TV-news frames[7]. Giving a source image, such as Figure VI.2, and an initial point, the sample points and the estimated Polygon region in first learning process is shown in Figure VI.3.

Apparently, the outliers distributed in the right part of image seriously affect the estimated Polygon descriptor.

Figure VI.4 shows the converged Polygon descriptor in the following learning processes which use the weighted sample points. Apparently, most outliers distributed in the right part of image are not covered by the estimated model now.

Figure VI.5 shows the final result of overall learning process, the last learning process corrects the number of normal vectors and fits the distribution region of data points with similar color.

3 Concluding Remarks

A cam-corder gets images of a 3D object by received the reflected light via lens. Since the received image is the projection of a 3D object, the polygon shape of this projection is thus useful to detect the movement of objects, including translation, rotation, spinning, leaving, and so on. For example, spinning a cube 90 degree will create a projection region in a hexagonal shape. By tracking the normal vector movement of a polygon descriptor, the 3D movement can therefore be recognized. However, 3D projection is much more complicate than that. For example, an object may be shadowed by another objects. Therefore, A 3D tracking application has to depend on the criteria of real applications.

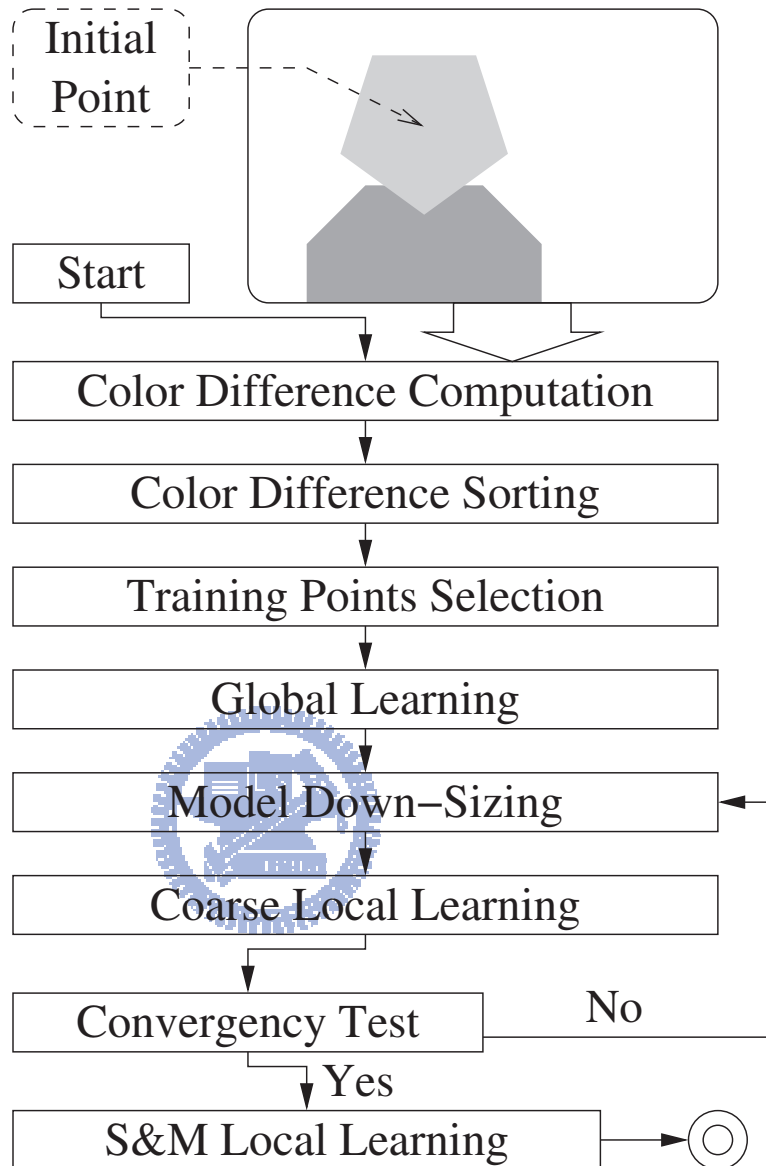


Figure VI.1: Flow chart of Polygon descriptor based region selection.



Figure VI.2: A sample image

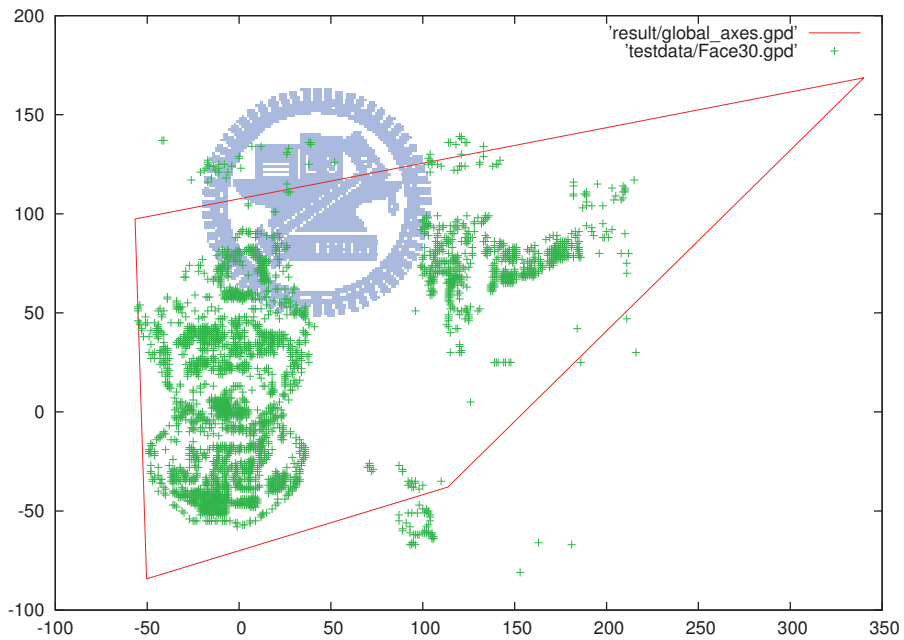


Figure VI.3: An example of sample points and the polygon region estimated in initial learning process.

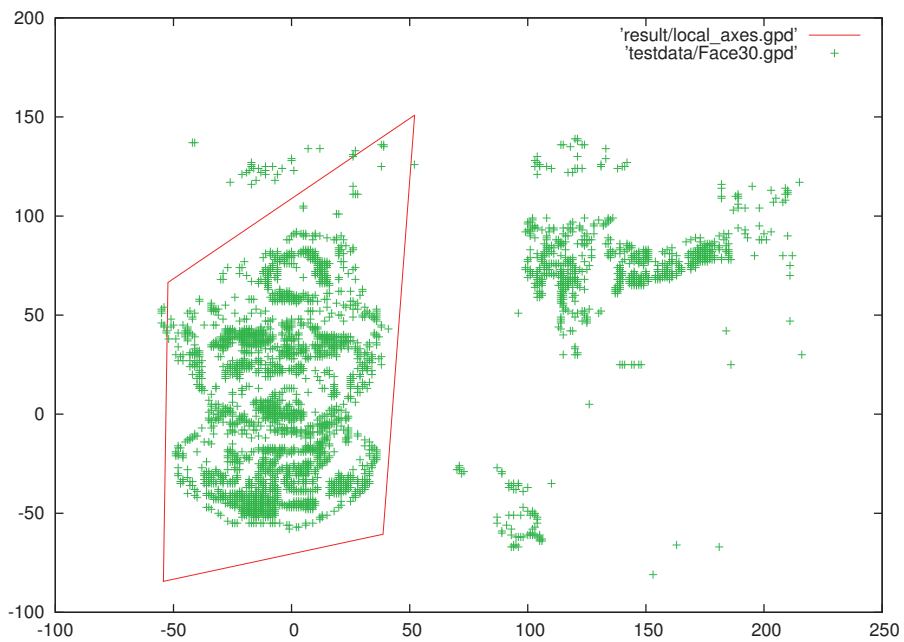


Figure VI.4: An example of sample points and the converged polygon region in the following learning processes.

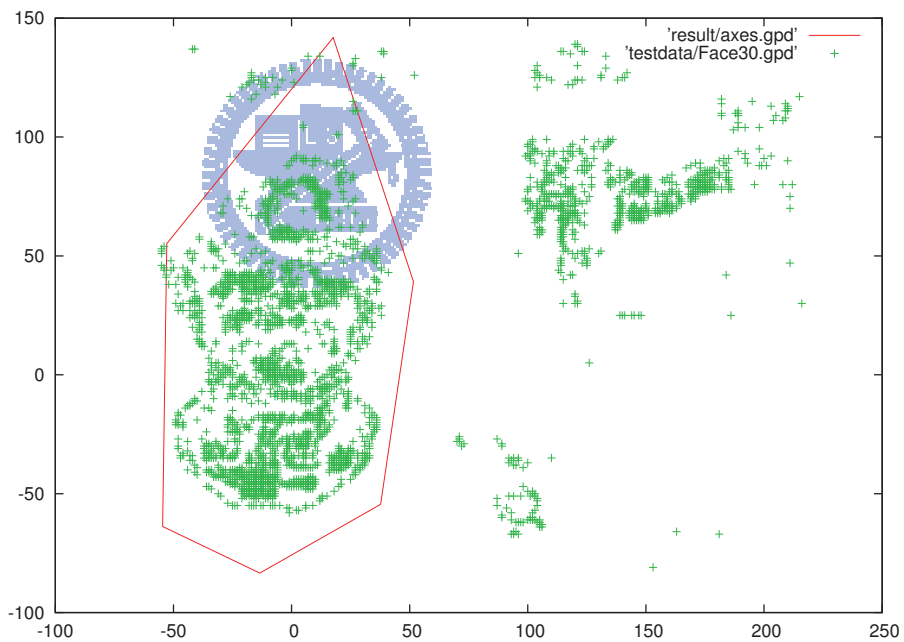


Figure VI.5: An example of the training pixels and the resulted polygon region.

Chapter VII

Conclusion

In this dissertation, Polygon descriptor, a polygon-based shape model, is proposed to represent a data distribution in numerical expressions. Three extensions of Polygon descriptor, 1) measurement of shape deformation, 2) virtual geometry for similarity based clustering, and 3) polygon based region representation, are exhibited to show that Polygon descriptor is a versatile model.

Since the shape of data distribution reflects the dependency among system factors and the dependency among system factors decides the behavior of a system, a shape model of data distribution captures the characteristics of system behavior. Therefore, by modeling the data distribution of signals, which is sampled from system factors, in numerical shape model, such as the proposed Polygon descriptor, data mining and pattern recognition methods can be used to mining information about the abstract system behavior.

To perform data mining applications for System behavior, the first extension, measurement of shape deformation, is proposed. Such similarity estimation can be useful because many data mining and pattern recognition methods, such as K Medoid, KNN (K Nearest Neighbor), binary search and so on, are based on the similarity values among data objects. For exam-

ple, based on the estimated similarity values, similarity index can be built to support the search function for data archives. Since similarity values are measured based on the difference between polygon descriptors, and a polygon descriptor reflects the status of system behavior, systems behavior can be recognized or analyzed by using the similarity measurement for Polygon descriptor. That is, abstract system behavior can be represented by a computable numerical expression by modeling the data distribution using the proposed Polygon descriptor and the similarity measurements of polygon descriptors. Based on the computable numerical expressions and the similarity values between them, abstract system behaviors can therefore be searched, recognized, analyzed and so on, by data mining, pattern recognition, and data analysis techniques.

A real world application is Financial data mining, by building the similarity index between every pair of data groups, such as periods, investors can search similar targets as their investment references. Or they can investigate or browse the change of market.

On the other hand, Polygon descriptor can also be used to improve the function of similarity based clustering. Generally, a clustering method divides the feature space into cells by creating decision boundaries. Decision boundaries/hyper-plane partition a space into (general) polygonal regions. The action of computing the variance for each clustering is like pulling or pushing the decision boundaries in between clusters. However, for similarity based clustering, feature space is usually not available. Thus, estimating variances is usually difficult. By using Polygon descriptor, the variance value can be estimated by an inverse process or similarity computation. Based on this method, a real world application, variable-sized thumbnail of web gallery, is demonstrated by using the variance enhanced K-Medoid method

to cluster images into groups according to the similarities between images.

The third extension of Polygon descriptor is to select or track the movement of a single color region. Describing region by a polygonal model has many advantages. The most significant advantage is that the vertices of a polygon can be used to track the movement of shape, such as spinning. For example, by modeling region in two consequent images using polygons, and mapping the vertices of these two polygon. The movement of each vertex show the movement of region and its transformation.

In summary, polygon is suitable to describe any shape. Most shapes can be approximated by polygons. Besides, many graphic theories depends on polygon based unit, such as mesh. The polygon shape region of data distribution also indicates the data dependencies among variants. And, polygons are friendly to computation and visualization. There are so many advantages to represent a shape by polygon regions. The proposed training algorithm for Polygon descriptor is based on only a few simple operations, such as cumulating and distributing. Since the algorithm is simply a great deal of executions of these simple operations, the training algorithm is potentially to be implemented by parallel computing. That is, the proposed method is intended to be simple to avoid trouble in computing power.

On the other hand, Polygon descriptor can be extended to various real world applications. Thus, Polygon descriptor is designed to be simple to be used without too much limitation. That why a Polygon descriptor describes a shape by normal vectors instead of exact coverage region. Normal vectors of polygon region boundaries reflects more information about the shape and can be covered to various form of shape descriptions.

Based on the simple, efficient, meaningful, and robust basis of Polygon descriptor, various of applications can be developed, for example, Independent

Component Analysis. The application of Polygon descriptor is not limited to the three extension demonstrations. Various kind of applications, such as searching, clustering, tracking, detection, recognition, and signal processing, can be achieved by extending the idea of Polygon descriptor. Besides, Polygon descriptor has the ability to creating much more unusual applications, such as high dimension data visualization, abstract feature extraction and so on.



Bibliography

- [1] The demonstration of variance enhanced k-medoid model. <http://www.csie.nctu.edu.tw/~pslai/askmddemo/>.
- [2] Taiwan economic journal data bank. <http://www.tej.com.tw/>.
- [3] Chang Wook Ahn and R. S. Ramakrishna. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transaction on Evolutionary Computation*, 6(6):566–579, December 2002.
- [4] Benoit Bellone and Erwan Gautier. Predicting economic downturns through a financial qualitative hidden markov model. Working Paper, available at <http://bellone.ensae.net/bellonepaper.html>.
- [5] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, ICSI-TR-97-021, University of Berkeley, 1997.
- [6] Gustavo Deco and Bernd Schurmann. *Information Dynamics*. Springer, 2001.
- [7] Hsin-Chia Fu, P.S. Lai, Lou R.S., and Pao H.-T. Face detection and eye localization by neural network based color segmentation. In *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 2, pages 507–516, 2000.

- [8] Keinosuke Fukunaga and David L. Kessell. Application of optimum error-reject functions. *IEEE Transactions on Information Theory*, 18(6):814–817, 1972.
- [9] A. Goshtasby. Description and discrimination of planar shapes using shape matrices. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 7:738–743, 1985.
- [10] Stephane Gregoir and Fabrice Lengart. Measuring the probability of a business cycle turning point by using a multivariate qualitative hidden markov model. *Journal of forecasting*, 19(2):81–102, March 2000.
- [11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning :Data Mining, Inference, and Prediction*, chapter 14.3.6, pages 468–470. Springer, 2001.
- [12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning :Data Mining, Inference, and Prediction*, chapter 14.3.10. Springer, 2001.
- [13] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- [14] Ruby L. Kennedy, Yuchun Lee, Benjamin Van Roy, Christopher D. Reed, and Dr. Richard P. Lippmann. *Solving Data Mining Problems through Pattern Recognition*, chapter 10.4.3, pages 1018–1023. Prentice Hall, 1998.

- [15] Ruby L. Kennedy, Yuchun Lee, Benjamin Van Roy, Christopher D. Reed, and Dr. Richard P. Lippmann. *Solving Data Mining Problems through Pattern Recognition*, chapter 10.4.9, pages 1050–1053. Prentice Hall, 1998.
- [16] Ruby L. Kennedy, Yuchun Lee, Benjamin Van Roy, Christopher D. Reed, and Dr. Richard P. Lippmann. *Solving Data Mining Problems through Pattern Recognition*, chapter 10.4.7, pages 1041–1047. Prentice Hall, 1998.
- [17] Por-Shen Lai and Hsin-Chia Fu. A polygon descriptor based similarity measurement of stock market behavior. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 806–812, Singapore, 2007.
- [18] Ping-Chen Lin and Jiah-Shing Chen. A genetic-based hybrid approach to corporate failure prediction. *International Journal of Electronic Finance*, 2(2):241–255, March 2008.
- [19] C. L. Liu. *Elements of Discrete Mathematics*, chapter 4, page 126. McGraw Hill, 1985.
- [20] Sven Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.
- [21] Udi Manber. *Introduction to Algorithms - A Creative Approach*, chapter 7.5, pages 201–208. Addison Wesley, 1989.
- [22] Udi Manber. *Introduction to Algorithms - A Creative Approach*, chapter 11.5.2, pages 365–368. Addison Wesley, 1989.
- [23] Udi Manber. *Introduction to Algorithms - A Creative Approach*, chapter 6.8, pages 155–158. Addison Wesley, 1989.

- [24] Udi Manber. *Introduction to Algorithms - A Creative Approach*. Addison Wesley, 1989.
- [25] Udi Manber. *Introduction to Algorithms - A Creative Approach*, chapter 7.10.2. Addison Wesley, 1989.
- [26] Hsiao-Tien Pao. Forecasting electricity market pricing using artificial neural networks. *Energy Conversion and Management*, 48(3):907–912, March 2007.
- [27] Time Terasvirta, Dick van Dijk, and Marcelo C. Medeiros. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting*, 21:755–774, 2005.
- [28] Robert A. Wagner and Michael J. Fischer. The string to string correction problem. *Journal of the ACM*, 21(1):168–173, January 1974.
- [29] Xiaoming Zhang and Paul L. Rosin. Superellipse fitting to partial data. *Pattern Recognition*, 36(3):743–752, 2003.



Curriculum Vitae

Por-Shen Lai graduated from National Chiao-Tung University at 1999, and got his master degree at 2001. Before starting his Ph.D program at 2002, he worked for Pen Power Technology for 1 year. He has join IAdea Corporation since 2009.

He started his research in Neural Network, Data Mining, and Statistical Learning methods since 1999, and creating methods to work on pattern recognition, and multimedia application. After 2001, the focus of his research became to create data model for financial utilities. He is also interesting in developing web-oriented applications and integrated systems, because of the need form his work.

