A Comparative Study of Data Clustering Techniques

Himanshu Mishra [#1], Shuchi [*2],Shashi Prakash Tripathi [$3]

Centre of Computer Education, Institute of Professional Studies, Science Faculty, University of Allahabad, India

---***---

Abstract-- Extraction of pattern from data and classifying them is very important for dealing with huge amount of data. Putting similar data into groups is called Data Clustering. Clustering algorithm partitions dataset in to various groups such that groups have similarities. This paper presents comparison between some common document clustering techniques. In particular, we compare the most representative offline clustering techniques: Fuzzy C means clustering, K-means clustering, Subtractive clustering, and Mountain clustering. The accuracy and performance of above four is tested on various aspects.

KEYWORDS: DATA CLUSTERING, FUZZY C MEANS CLUSTERING, K-MEANS CLUSTERING, SUBTRACTIVE CLUSTERING, MOUNTAIN CLUSTERING.

I. INTRODUCTION

Discovering groups and identifying interesting patterns is process of clustering. It is an effective approach for finding relations in data [1]. The concept of clustering, or data grouping is nearer to how human thinks and it is also very common in nature; we always try to summarize the data in to small groups whenever we got large amount of data.it helps to perform analysis over it. Moreover, generally the data that we have in many problems have natural groupings because of some intrinsic properties.it is really a tough process for human to recognizing that property and combined them in to a group because of that property.it is easy to recognize low dimensional data but hard to recognize the data that has higher dimensionality. This is why the computing methods came to reduce the human effort and make it more accurate in less time. The subject of the paper "Data Clustering Methods". The clustering algorithm is not important for categorizing the data but it also help for data compression and constructing different models.

The similarities in data helps to express the data by using fewer examples. If we are able to find how the data is grouped, then we can easily from a model of the problem based on those clusters.

In this paper, four of the most representative off-line clustering techniques are reviewed:

- K-means (or Hard C-means) Clustering,
- Fuzzy C-means Clustering,
- Mountain Clustering, and
- Subtractive Clustering.

These clustering techniques are generally used in conjunction with Fuzzy Modelling [2] and radial basis function networks (RBFNs). The results that are shown in this paper are comparative study of different techniques and the effect of different parameters in the process.

Earlier explained, data clustering is partitioning of a data set into several groups such that the similarity within a group is larger than that among groups. This implies that data have some inherent property; otherwise if we perform uniform distribution, then we can't find data clusters, or will lead to artificially introduced partitions.

There may be a problem of overlapping data groups. And because of this the efficiency of the clustering degraded, and this reduction is proportional to the amount of overlap between groupings. The techniques that are used in this paper are used in conjunction with other fuzzy models [3] or sophisticated neural. Specifically, these techniques can be considered as preprocessors for determining the initial locations for radial basis functions or fuzzy ifthen rules. We approach to find the cluster center to represent each cluster. A cluster center is a way to tell where the heart of each cluster is located, so that later when presented with an input vector, the system can tell which cluster this vector belongs to by measuring a similarity metric between the input vector and all the cluster centers, and finding the nearest cluster.

Some of the clustering techniques rely on knowing the number of clusters a priori. In that case the algorithm tries to partition the data into the given number of clusters. Fuzzy C-means and K-means clustering are of that type. In other cases knowing the number of clusters from beginning is not necessary; instead the algorithm starts by finding the clusters in sorted order like first large cluster, and then goes to find the second, and so on. Subtractive and Mountain clustering are of that type. In both cases a knowing clusters number is useful; however if the number of clusters is not known, K-means and Fuzzy C-means clustering cannot be used

II. BACKGROUND AND RELATED WORK

A. K-Means Algorithm

The k-means algorithm [Hartigan 1975; Hartigan & Wong 1979] is by far the most popular clustering tool used in industrial and scientific applications. The name comes from representing each of k clusters C by the mean (or weighted average) c of its points, the so-called centroid. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. The sum of discrepancies between a point and its centroid expressed through appropriate distance is used as the objective function. K-means clustering (or Hard C-means clustering, as compared to Fuzzy C-means clustering.) This technique has been applied to a variety of areas, including image and speech data compression, and with the help of radial basis function networks data preprocessing for system modeling, and task decomposition in heterogeneous neural network architectures [4]. This algorithm relies on finding cluster centers by trying to minimize a cost function of dissimilarity (or distance) measure.

A set of n vectors x_j, j= 1, 2, 3… n. are to be partitioned into c groups, G_i, i=1, 2… c. The cost function, based on the Euclidean distance between a vector x_k in group j and the corresponding cluster center c_i, can be defined by:

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\left(\sum_{k,\, \mathbf{x}_k \in G_i} \|\mathbf{x}_k - \mathbf{c}_i\|^2\right), \qquad (1)$$

Where is the cost Function within group i.

The partitioned groups are defined by a c*n binary membership matrix U, where the element u_{ij} is 1 if the jth data point x_j belongs to group i, and 0 otherwise. Once the cluster centers c_i are fixed, the minimizing u_{ij} for Equation (1) can be derived as follows:

$$u_{ij} = \begin{cases} 1 \text{ if } \|\mathbf{x}_j - \mathbf{c}_i\|^2 \le \|\mathbf{x}_j - \mathbf{c}_k\|^2, \text{ for each } k \ne i, \\ 0 \text{ otherwise.} \end{cases} \qquad (2)$$

Which means that x_j belongs to group i if c_i is the closest center among all centers. On the other hand, if the

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m}, \qquad (6)$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}}. \qquad (7)$$

membership matrix is fixed, i.e. if u_{ij} is fixed, then the optimal center c_i that minimize Equation (1) is the mean of all vectors in group i:

$$\mathbf{c}_i = \frac{1}{|G_i|} \sum_{k,\, \mathbf{x}_k \in G_i} \mathbf{x}_k, \qquad (3)$$

Where | G_i | is the size of G_i, or $|\, Gi \,| = \sum_{j=1}^{n} Uij$.

The algorithm is presented with a data set x_i, i= 1, 2, 3, …., n; it then determines the cluster centers c_i and the membership matrix U iteratively using the following steps:

Step 1: Initialization of cluster center C_i, ranging from 1 to C.

Select C randomly from data sets.

Step 2: Solve Equation (2) and find the membership matrix U.

Step 3: Using equation (1) compute the cost function. Stop if it is less than the certain tolerance value or its increment is comparatively below over a certain threshold.

Step 4: Update the values of cluster center using equation (3). Go to step 2.

The initial position of clusters centers a performance factor of K-Means algorithm, thus it is preferable to run the algorithm more times, each with a different set of initial cluster centers.

B. Fuzzy C-means Clustering

It relies on the basic idea of Hard C-means clustering (HCM), there is a difference that every data point of Fuzzy C- means belongs to a cluster of membership grade whereas the Hard C-Means may or may not be belongs to a certain cluster. The membership matrix is allowed to store binary values 0 and 1. However

$$\sum_{i=1}^{c} Uij = 1 \quad ; \qquad (4)$$

For every j belongs to 1 to n.

The cost function of Fuzzy C-means Clustering is generalization of equation (1):

$$J(U, \mathbf{c}_1, \ldots, \mathbf{c}_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2, \qquad (5)$$

Where u_{ij} € (1,n).c_i is the cluster center of fuzzy group d_{ij} = $||c_i - c_j||$ is the Euclidean distance between the ith cluster center and the jth data point; and m ∈ [1,∞) is a weighting exponent. The necessary conditions for Equation (5) to reach its minimum are

Above two condition is required for iteration of the algorithm. Fuzzy C-means Clustering determines the C_i and the membership matrix U using the following steps in batch operation mode:

Step 1: Find the value between 0 to 1 that satisfies the equation (4) and put it in membership matrix U.

Step 2: Calculate c fuzzy cluster centers c_i where i ranges 1 to c, using Equation (6).

Step 3: Compute the cost function according to Equation (5).

Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold.

Step 4: Compute a new U using Equation (7). Go to step 2.

Same as earlier stated for K- means performance. It is also advisable to start iteration with initial values of membership matrix U.

C. Mountain Clustering

It is another approach for finding cluster center based on a density measure called the mountain function. It is an approximation method of finding cluster center, hence it can be used as preprocessor for other sophisticated clustering methods. Firstly we form a grid on data space, the intersection of these grid lines potential to cluster center, represented as a set V. Then we generate a mountain function representing a data density measure. The height of the mountain function at a point v € V is equal to

$$m(\mathbf{v}) = \sum_{i=1}^{N} \exp\left(-\frac{\|\mathbf{v}-\mathbf{x}_i\|^2}{2\sigma^2}\right), \qquad (8)$$

Where x_i is the ith data point and σ is an application specific constant.

D. Subtractive Clustering

The computation growth exponentially in mountain clustering with the dimension of the problem; that is because we are evaluating the mountain function at every grid point. We can resolve it by using Subtractive Clustering because this technique uses data points as candidate for clusters centers, instead of grid points. That reduces the computation to problem size from problem dimension. However, the actual cluster centers are not necessarily located at one of the data points, but in most cases it is a good approximation, especially with the reduced computation this approach introduces.

Since every data point is a candidate for cluster centers, a density measure at data point x_i is defined as:

$$D_i = \sum_{j=1}^{n} \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{(r_a/2)^2}\right), \qquad (10)$$

Where r_a is a positive constant representing a neighborhood radius. Hence, a data point will have a high density value if it has many neighboring data points.

The first cluster center Xc_1 is chosen as the point having the largest density value D_{c1}. Next, the density measure of each data point X_i is revised as follows:

$$D_i = D_i - D_{c_1} \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_{c_1}\|^2}{(r_b/2)^2}\right) \qquad (11)$$

Where r_b is a positive constant which defines a neighborhood that has measurable reductions in density measure. Therefore, the data points near the first cluster center X_{c1} will have significantly reduced density measure.

After revising the density function, the next cluster center is selected as the point having the greatest density value. This process continues until a sufficient number of clusters is attainted.

III.A COMPARATIVE STUDY OF DATA CLUSTERING TECHNIQUES

Earlier we have explained different clustering with their basic mathematical foundations. This section involves the comparison of the above explained techniques, and testing each one of them on a dataset. Firstly we portioned the dataset into two data sets: 2/3 the data for training, and 1/3 for evaluation. We divide the whole dataset in two cluster; Because of higher dimension visual representation is not possible; we mainly focus on performance measures rather than visual approaches.

Euclidean distance: similarity between input vector element and a cluster center. Since similarity metrics are sensitive on large ranges, every input variables must be normalized to within the unit interval [0, 1].

A. K-means Clustering

Performance measure	Test Runs									
	1	2	3	4	5	6	7	8	9	10
No. of iterations	7	6	9	6	5	5	3	7	11	7
RMSE	0.469	0.469	0.447	0.469	0.632	0.692	0.692	0.447	0.447	0.469
Accuracy	78.0%	78.0%	80.0%	78.0%	60.0%	52.0%	52.0%	80.0%	80.0%	78.0%
Regression Line Slope	0.564	0.564	0.6	0.564	0.387	0.066	0.057	0.6	0.6	0.564

K-means Clustering Performance Results

In K-means clustering we try to find cluster center by minimizing the cost function J. By using Equations (2) and (3) we update the membership matrix and cluster center, respectively, till no further increment noticed in cost function . Because the algorithm initializes the cluster center randomly, this affects the performance. That's why it is advisable to have multiple test runs. Accuracy of algorithm can be realized by testing the evaluation sets. After the determination of cluster center, the evaluation vector assigned to their corresponding cluster center on the basis of Euclidian distance. Then an error

Measure is then calculated; we use the root mean square error (RMSE) is used for this purpose. We also calculate the accuracy percentage. We iterate the algorithm 10 times for better results. K- Means performance result table lists the results of those runs.

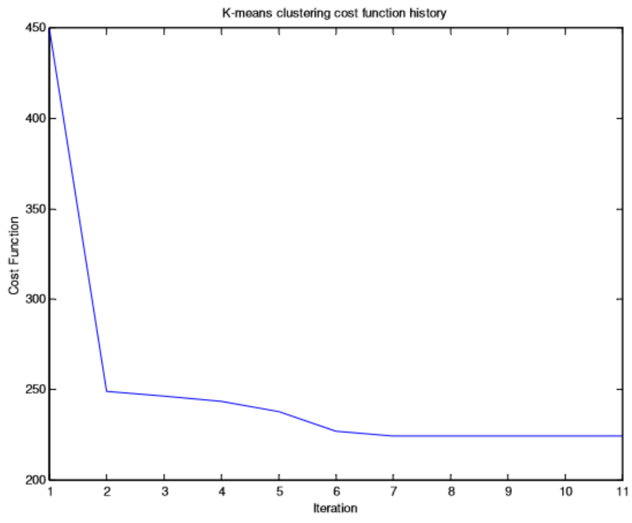


Figure 1. K-means clustering cost function plot

Figure 1 shows the plot of costa function over time.

We performed a regression analysis resultant clustering against the original classification. For better performance the slope should be nearer to 1. Figure 2 shows the regression analysis of the best test case.

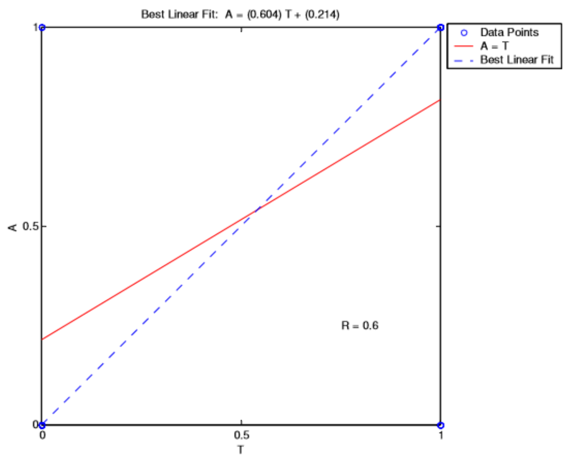


Figure 2. Regression Analysis of K-means Clustering

B. Fuzzy C-means Clustering

In Fuzzy C-means clustering every data point can have different degrees of membership to their respective clusters; thus it helps to eliminate the effect of hard membership. This approach keep occupying the fuzzy

Measures as the basis for U (membership matrix) calculation and identification of cluster center.

Fuzzy C-means Clustering starts by assigning random values to the U, thus for getting higher probability of good results we apply it several times. However, there is no variance in accuracy and performance when algorithm tested several times. For testing the results, every vector in the evaluation data set is assigned to one of the clusters with a certain degree of belongingness (as done in the training set). However, because the output values is in binary (either 1 or 0), the evaluation set degrees of membership are defuzzified to be tested against the actual outputs.

Performance measure	Weighting exponent *m*							
	1.1	1.2	1.5	2	3	5	8	12
No. of iterations	18	19	17	19	26	29	33	36
RMSE	0.469	0.469	0.479	0.469	0.458	0.479	0.479	0.479
Accuracy	78.0%	78.0%	77.0%	78.0%	79.0%	77.0%	77%	77%
Regression Line Slope	0.559	0.559	0.539	0.559	0.579	0.539	0.539	0.539

Fuzzy C-means Clustering Performance Results

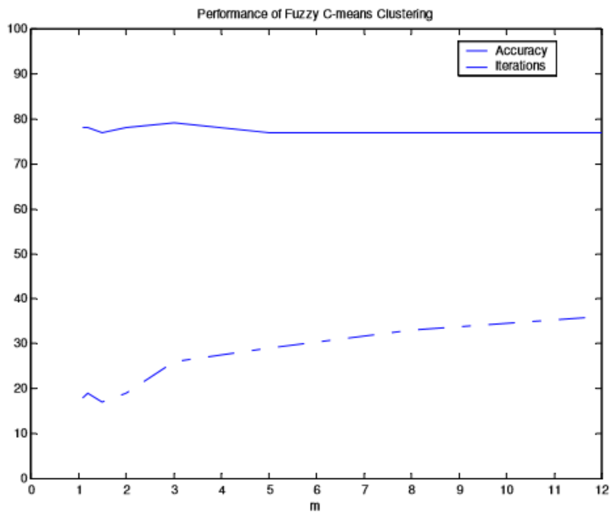


Figure 3. Fuzzy C-means Clustering Performance

This data clustering has good accuracy and requires less number of iterations. Figure 3 shows the accuracy and number of iterations against the weighting factor.

In general, the Fuzzy C-means Clustering technique Not good for the selected problem it performs as k- means algorithm. Both showed close accuracy; moreover Fuzzy C-means Clustering is slower because of fuzzy calculations.

C. Mountain Clustering

In this clustering technique we divide data space in to grid points and then calculate the mountain function at every grid point. Mountain function represents the density of the data point. The performance of the Mountain clustering is gravely affected by the dimension of the problem. The computation rises exponentially because the mountain function is calculated at every grid point. For a problem with c clusters, k dimensions, j data points, and a grid size of t per dimension, the required number of calculations is:

Performance measure	Test Runs									
	1	2	3	4	5	6	7	8	9	10
RMSE	0.567	0.469	0.567	0.491	0.549	0.567	0.567	0.529	0.693	0.469
Accuracy	68.0%	78.0%	68.0%	76.0%	70.0%	68.0%	68.0%	72.0%	52.0%	78.0%
Regression Line Slope	0.351	0.556	0.344	0.515	0.429	0.346	0.346	0.492	0.027	0.551

Mountain Clustering Performance Results

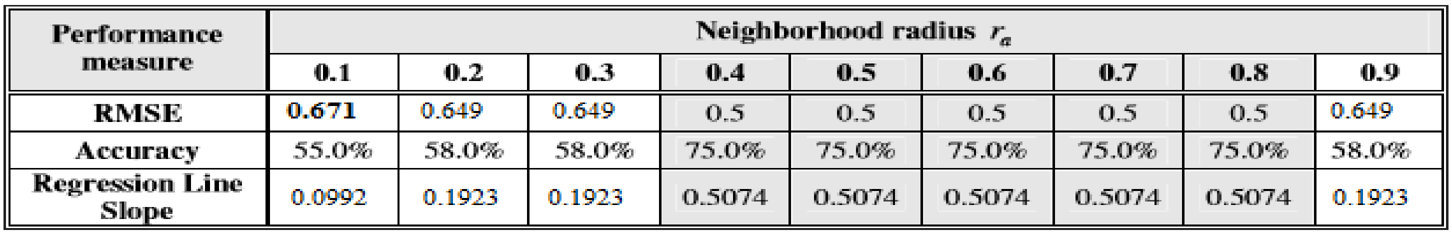
$$N = j * t^k + (c-1) * t^k \qquad (12)$$

So for any problem, with input data of 14 dimensions, 210 training inputs, and a grid size of 10 per dimension, the required number of mountain function calculation is approximately $2.01 * 10^{15}$ calculations.

D. Subtractive Clustering

It is very similar to the mountain clustering, there is a bit difference that we calculate density function at every data point only. And the data points itself are the candidate for the cluster centers. This reduces the computation and making it possible in linear time whereas in case of mountain it was exponential. For a problem of c clusters and j data points, the required number of calculations is:

$$N = j^2 + (c - 1) j$$



Performance measure	Neighborhood radius r_a								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RMSE	0.671	0.649	0.649	0.5	0.5	0.5	0.5	0.5	0.649
Accuracy	55.0%	58.0%	58.0%	75.0%	75.0%	75.0%	75.0%	75.0%	58.0%
Regression Line Slope	0.0992	0.1923	0.1923	0.5074	0.5074	0.5074	0.5074	0.5074	0.1923

Subtractive Clustering Performance Results

This algorithm does not rely on any randomness, so the results are fixed. However, we can test the effect of the two variables r_a and r_b on the accuracy of the algorithm.

Those variables represent a radius of neighbourhood after which the effect (or contribution) of other data points to the density function is diminished. Usually the b r variable is taken to be as 1.5 r_a..

We can not take r_a very large or small t is clear from the plot because if we take it tto small then the density function will not going to effect on neighbouring data Points [5] ; if we take it very large then the density function will affected by the neighbouring data points. So r_a should contain a value between 0.4 to 0.7[6][7].

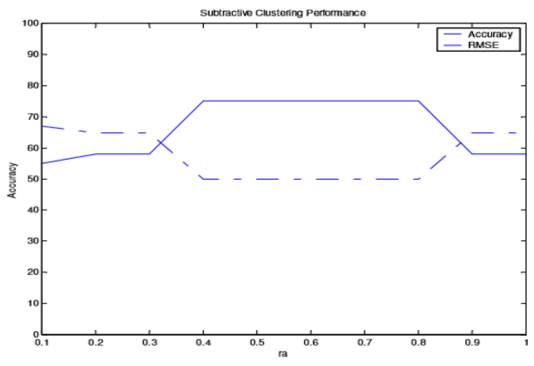


Figure 4. Subtractive Clustering Performance

As summary best achieved from the above study is stated below in the form of performance chart.

We have earlier explained the four data clustering techniques. Let's see the comparison:

Algorithms	**Comparison Aspect**			
	RMSE	Accuracy	Regression Line Slope	Time (sec)
K-Means	0.445	80.0%	0.62	0.9
FCM	0.468	78.0%	0.553	2.21
Mountain	0.470	78.0%	0.557	118.1
Subtractive	0.52	75.0%	0.5073	3.61

From this comparison we can conclude some remarks:

• K-means clustering has higher accuracy, lower RMSE and required less time.

• Mountain required huge time for large calculation.

• Mountain clustering is suitable for less dimension data sets.

•Fuzzy c- means is closer to k means but more time to calculate fuzzy functions [8][9].

• In subtractive clustering, one has to very careful for selecting the value of the neighbourhood radius r_a.

•Since none of the algorithm is best suited for the problem of higher dimension with overlapping in some of the dimension.

IV. CONCLUSION AND FUTURE WORK

We have reviewed four data clustering technique in this paper, namely: Fuzzy C-means clustering, K-means clustering, Subtractive clustering and Mountain clustering. The approaches solves the problem by portioning the data set in to cluster based on some similarity measures. The four methods have been implemented and tested against a data set .The problem presented has a high number of dimensions, which might involve some complicated relationships between the variables in the input data.

We will try to give an algorithm for clustering that have better accuracy and performance for higher dimension data sets.

REFERENCES

[1]. Khaled Hammouda Prof. Fakhreddine Karray University of Waterloo, Ontario, Canada, "A COMPARATIVE STUDY OF DATA CLUSTERING TECHNIQUES"

[2]. Jang, J.-S. R., Sun, C.-T., Mizutani, E., "NeuroFuzzy and Soft Computing – A Computational Approach to Learning and Machine Intelligence," Prentice Hall.

[3]. Azuaje, F., Dubitzky, W., Black, N., Adamson, K., "Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach," IEEE Transactions on Systems, Man, and CyberneticsPart B: Cybernetics, Vol. 30, No. 3, June 2000 (pp. 448)

[4]. Lin, C., Lee, C., "Neural Fuzzy Systems," Prentice Hall, NJ, 1996

[5]. The MathWorks, Inc., "Fuzzy Logic Toolbox – For Use With MATLAB," The MathWorks, Inc., 1999.

[6]. Tulika Narang and Prof. R. R. Tewari "Multilevel Approach to Ontology Driven Clustering of Web Documents", in:Proceedings of the International Conference on Information and Knowledge Engineering , IKE'12,July 2012, USA, ISBN: 160132-222-4

[7]. Tulika Narang and Prof. R.R. Tewari" Som Based Clustering of Web documents using an Ontology" ,International Journal of Engineering Research and Science and Technology,Vol 2, No. 3, August 2013,ISSN:2319-5991

[8]. Tulika Narang, "Hiearchical clustering of documents:A brief study and implementation in Matlab", in:Proceedings of the International Conference on Emerging Trends of Engineering, Science, Management and Its Applications,March 2015,New Delhi,India

[9]. Tulika Narang and Dharmendra.K. Yadav"Capturing dynamic behavior in Relational model", in: Proceedings of International conference on Software Engineering and Research Practices,July 2011,USA