Beyond Kmedoids: Sparse Model Based Medoids Algorithm for Representative Selection

Yu Wang, Sheng Tang, Feidie Liang, YaLin Zhang, and Jintao Li

Advanced Computing Research Laboratory, Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{wangyu,ts,liangfeidie,zhangyalin,jtli}@ict.ac.cn

Abstract. We consider the problem of seeking representative subset of dataset, which can efficiently serve as the condensed view of the entire dataset. The Kmedoids algorithm is a commonly used unsupervised method, which selects center points as representatives. Those center points are mainly located in high density areas and surrounded by other data points. However, boundary points in the low density areas, which are useful for classification problem, are usually overlooked. In this paper we propose a sparse model based medoids algorithm (Smedoids) which aims to learn a special dictionary. Each column of this dictionary is a representative data point from the dataset, and each data point of the dataset can be described well by a linear combination of the columns of this dictionary. In this way, center and boundary points are all selected as representatives. Experiments evaluate the performances of our method for finding representatives of real datasets on the image and video summarization problem and the multi-class classification problem, and our method is shown to out-perform state-of-the-art in accuracy.

Keywords: representative subset, sparse model, dictionary learning.

1 Introduction

In the field of machine learning, computing vision and information retrieval, the scale of dataset grows at an ever increasing rate. Dealing with massive dataset is time- and memory- consuming. Thus being able to select a relatively small number of samples from a dataset, which can serve as a condensed view of the entire dataset, is of importance. Using those representative samples for classification and clustering algorithms can greatly reduce the memory requirement and computational time. In addition, representative samples can be available for online extension.

Kmedoids [1] is a common unsupervised method which produces representative samples. Similar to Kmeans, it assumes data points are distributed around several cluster centers. But unlike Kmeans, those cluster centers of Kmedoids are data points themselves, called medoids. Those medoids are usually located in the high density areas, ignoring the low density boundary zones. As shown in Fig.1 medoids of Kmedoids algorithm are mostly concentrated in high density areas of the distribution of the original dataset. But for classification problem, e.g. SVM, low density boundary areas deserves more concern [2].

S. Li et al. (Eds.): MMM 2013, Part II, LNCS 7733, pp. 239–250, 2013.

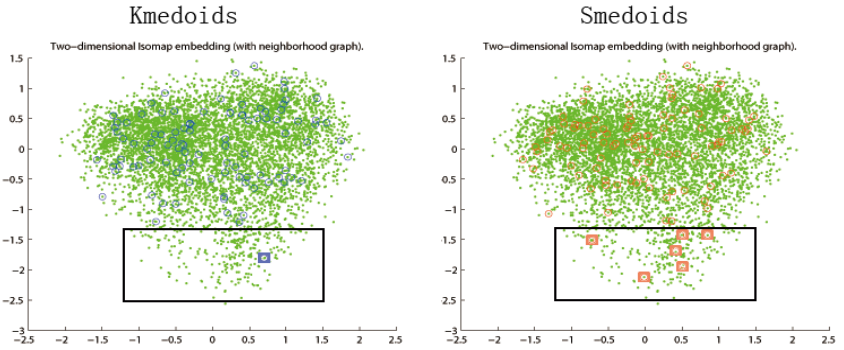

Fig. 1. Visual comparison of Smedoids and Kmedoids on digit '2' of USPS. The green points represent samples of digit '2'. The points circled by red are representative points selected by our method, while those circled by blue are selected by Kmedoids. The representative points of Kmedoids are mostly concentrated in the high density areas of the entire distribution, e.g. there is only one representative point in the low density areas (in the black box); While ours has many representatives in the low density zones, e.g. six points are selected in the black box.

In order to get accurate condensed view of the entire dataset, we propose a sparse model based medoids algorithm (Smedoids), to find a compact dictionary whose columns are representative samples. Unlike Kmedoids, in which each data point is assigned to one center reprehensive point, in our method each data point can be expressed by a linear combination of the representative points. In other words, representatives are subset of the entire data, which are referred most frequently by all the other data. In this way, whole distribution of original dataset can be well covered and deviation of our method between original data and representatives are less than Kmedoids'. As shown in Fig.1 representative points in our method are not necessarily the cluster centers and many are in the low density zones.

2 Related Work

This work is closely related to representative selection methods and dictionary learning methods. First, we briefly review some related representative selection methods. Then dictionary learning methods are introduced.

2.1 Representative Selection Methods

Several methods have been proposed for representative selection. Kmedoids [1] and Affinity propagation [3] are common unsupervised methods. Kmedoids, similar to Kmeans, is an iterative algorithm to find data centers surrounded by other data. When similarities between pairs of samples are given, Affinity propagation uses a message passing algorithm to find those data centers. When data are assumed to be low-rank, there are many methods [4],[5] using matrix factorization to select a few columns from the data matrix. The methods proposed by [7], [8] are recent supervised methods

for representative selection. The method by [7] aims to find representatives close to their class and far from other classes. While the method by [8] suggests a representative selection technique focused on having a large hypothesis margin to improve the performance of the 1-NN rule.

Our method is unsupervised, which is different from the above methods for the following reasons: First, representative points are not necessarily the centers as Kmedoids. In our method each data point can be described by a linear combination of representatives. Those representatives are those data points referred most frequently by other data in the dataset. Second, in the low density areas the number of representative points is more than those center methods. We illustrate that lots of points are near in high density areas. Because they are similar, they have high probability to refer the same representatives within acceptable range of error. Thus representatives in high density areas are highly reused. While in the low density areas, data points have larger distance than those high density points, so they have to refer their own neighbor points, and many representatives must be chosen in this zone. Third, our method uses dictionary learning method instead of the matrix decomposition approach, so it does not require the data to be low-rank.

Worth noting that [9] proposes a sparse modeling representative selection method (SMRS) for finding representative objects in two steps: it first uses all data points as dictionary for sparse coding and second selects representatives according to their sparse representations. But using dataset as a large redundant and coherent dictionary makes sparse coding unstable and expensive, while learning a compact dictionary can overcome those problems [11]. Our method is a dictionary learning method, which is stable and efficient. Besides we have compared with SMRS in experiments and received even better results.

2.2 Dictionary Learning Methods

Learning a dictionary from data under some constraints is widely used in computer vision and machine learning problems [6]. K-SVD algorithm [10] uses SVD decomposition of the error matrix to learn over complete dictionary from redundancy signals. Dictionaries according to many classes are constructed for clustering problem [11],[12]. Transfer learning task builds a common dictionary to find new features [13]. Task driven dictionary learning algorithm for classification is proposed by [14]. The method proposed by [15] uses online optimization based on stochastic approximation which is suitable for large-scale task.

However atoms of those dictionaries are not the original data points, hence they can't be used as representative points directly. Different from those previous works, the proposed Smedoids algorithm learns a dictionary which is subset of the dataset. Each data point in the dataset can be described as a linear combination of atoms from this learned dictionary, so this dictionary can be considered as the condensed view of the dataset.

3 Problem Formulation

Consider a set of data points in R^m arranged as the columns in data matrix $X = \{x_1, ...x_n\}$, the representative selection methods seek the representative matrix $D = \{d_1, ...d_l\}$ which are subset of X and the condense view of the original dataset.

The Kmedoids algorithm learns representative points D in two steps: first dividing data points into k parts, and each data point has only one representative in D; Second fixing the division and finding a new medoids in each part. The problem is formulated as follows:

$$\min_{a} \sum_{i=1}^{n} \| x_i - Da_i \|_2^2, s.t. \| a_i \|_0 = 1 \ , \tag{1}$$

$$\min_{D} \sum_{i=1}^{n} \| x_i - Da_i \|_2^2 \ , \tag{2}$$

Where a_i is the coefficient of x_i. The ℓ_0-norm of a_i in Eq. (1) is to ensure that each x has only one representative point in D, and Eq. (2) solves a better D when fixing a_i. Eq. (1) and Eq. (2) can be rewritten as follow:

$$\min_{D,a} \sum_{i=1}^{n} \| x_i - Da_i \|_2^2, s.t. \| a_i \|_0 = 1 \tag{3}$$

ℓ_0-norm of a_i equals 1 constraint the solution to be center points in high density areas, but as pointed out in [2] high density areas is weak for classification. Thus in our proposed Smedoids algorithm, we extend $\| a_i \|_0 \le s$, where s is the maximum number of nonzero items in the sparse representations, and the problem turns to be sparse modeling [16]. Since the ℓ_0 norm is NP-hard, we replace the ℓ_0 norm with the ℓ_1 norm. Different from the former works, the atoms of dictionary we get are the actual data points, which can describe the original data set. The formulation is following:

$$\min_{D,a} \sum_{i=1}^{n} \| x_i - Da_i \|_2^2 \quad \text{s.t. } \| a_i \|_1 \le s, D \in X \tag{4}$$

With our method the experiments show that the distribution of the dataset can be well covered by atoms of this dictionary, and the total reconstruction error $\| x_i - Da_i \|_2^2$ is less than the Kmedoids.

3.1 Smedoids Algorithm

The Smedoids algorithm we proposed is aim to solve Eq. (4). The Eq. (4) has two variables and is not a convex problem. We can fix one variable, then the Eq. (4)

becomes a convex problem. The Smedoids performs the following two steps iteratively: first learning sparse representation of each data point using LASSO algorithm [17]; Second fixing sparse representations and finding a better D column by column. By those steps the total reconstruction error of Eq. (4) is reduced gradually as shown in Fig. 2. Details are shown in Algorithm 1.

Algorithm 1. Smedoids

Input: $X \in R^{m*n}$, T (the number of iterations), l (the number of atoms in D), s (the maximal number of nonzero atoms in each sparse representation).

Output: $D \in R^{m*l}$.

Initialization: ℓ_2-normalize X, and randomly select l samples from X to initialize D.

Repeated until T

- Sparse coding: computing sparse representations $A = \{a_1, ..., a_n\}$ using LASSO by solving:

$$\min_{a} \sum_{i=1}^{n} \|x_i - Da_i\|_2^2 \text{ s.t.} \| a_i \|_1 \le s, D \in X \tag{5}$$

- Solving a better D: for the k-th column in D, fixing other columns:
  - Define the group of data X_{ref}, and $ref = \{i \mid 1 \le i \le n, a_i^k \ne 0\}$.
  - Define the sparse representation matrix A_{ref}, let $a_k = A_{ref}(k,:)$, $A_{ref}(k,:) = 0$.
  - Compute the error matrix E_k, by $E_k = X_{ref} - DA_{ref}$.
  - The total error function $g(d_k) = \| E_k - d_k a_k \|_2^2$.
  - For each $q \in ref$:

$$d_k = \min_x g(x_q) = \min_x \| E_k - x_q a_k \|_2^2 \text{ s.t. } a_k = x_q \setminus E_k. \tag{6}$$

The Smedoids algorithm is different from standard sparse modeling in the process of dictionary updating. Smedoids algorithm aims to select representatives which coincide with original data distribution. However standard sparse modeling method solves dictionary updating by 'calculating', which mixes the data points, so normally the dictionary is not consistent with original data distribution. In Algorithm 1 the dictionary is updated column by column. When updating the k-th column, other columns are fixed. The variable *ref* indicates the data points referred the k-th column, and error $g(d_k) = \| E_k - d_k a_k \|_2^2$ reflects the total error the current column causes. The next column should be the data point which can reduce g as Eq. (6). Updating corresponding a_k with d_k is suggested in [10] as an efficient implementation.

3.2 Convergence Analysis

Algorithm 1 has two steps, and in this section we prove that each step of algorithm 1 is convergent, so this algorithm is convergent.

The first step is to calculate the sparse representations of samples. For small s compared to n, the LASSO algorithm can robustly approximate the solution of Eq. (5) [10]. This assumption is natural in applications of image and video processing. Thus with fixed D this step decreases the solution of Eq. (4).

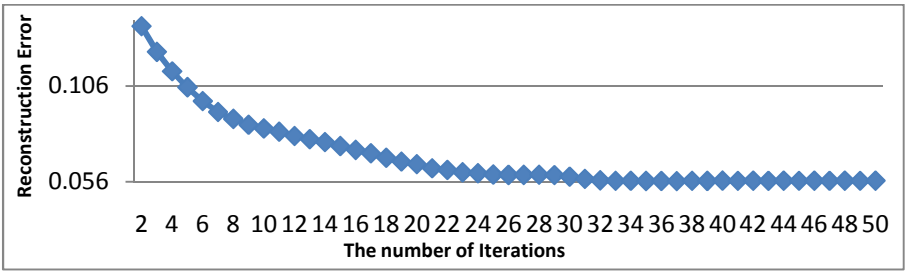

Fig. 2. Visualization of the convergence of the algorithm 1, the reconstruction error of Eq. (4) reduces gradually within 50 iterations

The second step is to find a better dictionary, when reduction or no change of the solution in Eq. (6) and the sparsity constraint are guaranteed, this step is convergent. Because the corresponding data is selected at first, the sparsity constraint is met [10]. The following proves that reduction or no change of Eq. (6)'s solution is guaranteed.

Theorem 3.1. Algorithm 1 can reduce or not change the error of the previous iteration for optimizing dictionary.

Proof: Finding a better dictionary can be rewritten as following:

$$f(D) = \min_{D} \| X - DA \|_2^2, \quad \text{s.t. } D \in X \tag{7}$$

Since this convex optimization problem admits separable constraints in the columns [15], updating the columns of dictionary one by one guarantees the convergence to a global optimum. When updating k-th column, the problem is following:

$$\begin{aligned} f(d_k) &= \min \| X - DA \|_2^2 = \min \| X_{nref} - DA_{nref} + X_{ref} - DA_{ref} - d_k a_k \|_2^2 \\ &= \min \| X_{nref} - DA_{nref} + E_k - d_k a_k \| \end{aligned}, \tag{8}$$

where ref indicates those samples which refer the k-th column, nref indicates those don't. We set the k-th row of the matrix to zero: $A_{ref}(k,:) = 0$. In the above equation, the $(X_{nref} - DA_{nref})$ is constant since other columns are fixed, so $g(d_k) = \| E_k - d_k a_k \|_2^2$ is the total error caused by the k-th column.

For $d_k \in X$, there is a data point x_j ,where $d_k = x_j$ and $x_j \in X_{ref}$. If no other data points minimize g , x_j is still selected by LASSO algorithm and the solution does not change; otherwise, other better data point is chosen to reduce the error.

From the above proof, we can conclude that algorithm 1 is convergence, so with finite iterative number T the optimized dictionary is always solved. In each iterative the complex of Lasso algorithm is $O(lmsn + ls^2 n)$, and the dictionary updating is $O(ln_{ref})$. In fact on many-core platform [18], the parallelization of dictionary updating reduces the complexity to $O(n_{ref})$. Thus the total complexity of the algorithm is $T*(O(lmsn + ls^2 n) + O(ln_{ref}))$. When $n > 2Tls, m \gg s$, the upper bound of this complexity is $O(mn^2)$.

Fig. 3. Visualization of the dictionary after updating with new added data, the first dictionary (on the left) is training on some samples of digit '1'. Then this dictionary is used as an initialization dictionary for new adding samples. The second dictionary (on the right) has learned some new atoms marked in red while other atoms are almost consistent with the first one.

3.3 Online Extension

Since the representative points coincide with original data, they can be reused when new samples in the same or similar classes are added. Let D be the representative points solved in the current dataset and X_{new} be the new added data. There are two extension methods. Let D be the initialization of the new dictionary, e.g. as shown in Fig. 3 or combine D and X_{new} into a new data collection.

When adding a new representative dictionary D_{new} of other class, since D_{new} is sufficient to describe the data collection of the other class, the optimization dictionary can be achieved simply by combining $[D\ D_{new}]$.

4 Experiments

In this section, we illustrate image and video summarization and multi-class classification problem for evaluating the performances of our method for finding representatives of real datasets.

4.1 Image and Video Summarization

We demonstrate that those representatives selected by the proposed algorithm can well summarize image and video datasets, so this algorithm can be used as a preprocess step in applications [24, 25].

First we consider the summarization of images of USPS dataset. The dataset consists of different variation of ten digit characters. The representatives of USPS are shown in Fig. 4 and it is worth notice that some marked representatives are not the center samples. Those marked samples have fewer occurrences than others and are hard to be classified, e.g. the marked representative of digit 4 and digit 9, digit 1 and digit 2. Those boundary representatives are useful for marginal decision in SVM training process.

Fig. 4. Representatives selected by our algorithm for the images of USPS dataset. Those representatives stand for different variation of each digit.

Fig. 5. Frames selected by our method for a one-shot video. Nine representative frames selected by our algorithm summary the activities of the video as follows: (1) a man stands by a window;(2) a man talks to someone across the widow;(3) a woman and a man enter the room;(4) one man leaves the room;(5) the first man sitting with the woman takes away her crown and hands out the window;(6) the man leaves the room; (7) the woman see the thief outside the window taking her crown;(8) the thief runs; (9) the woman passes out on the sofa.

Fig. 6. Frames selected by our method for a multi-shot video. Different numbers of representative frames are automatically computed according the amount activities of each shot as: one representative frame for the 2-th shot; three representatives for the 3-th shot; two representatives for the 4-th shot; two representatives for the 5-th shot.

Next we chose a 1,536-frame one-shot video and a 782-frame multi-shot from [19]. The one-shot video contains continuous actives in a fixed background. We use algorithm 1 to extract nine representative frames. As shown in Fig. 5 those representatives well cover the whole activities of this video. This result is better than [9] which have missed the frame where a man is passing the crown to the window. The 5-shot video contains 4 different scenes, including cartoon, sky, virtual scene and rocket launch. We extract eight representative frames. Those representatives capture all those scenes but not all those shots as shown in Fig. 6. It's noteworthy that 1-th shot is similar with 5-th shot regardless background or activity, so they share representatives, that's why 1-th shot has no representatives. Different number of representatives reflects the number of activities in each shot. Two kids are throwing a ball in the third shot. Three representatives capture the main activities of this event: (1) a boy is ready to throw a ball; (2) a girl receives the ball; (3) the girl holds the ball. Relatively static shot has fewer representatives as 2-th shot.

4.2 Classification Performance Using Representatives

We now evaluate the performance of our method as well as other algorithms for finding representatives that are used for multi-class classification problem. For training set in each class we only select a few representatives and use them as a reduced training dataset. It is believed that the better those representatives condense the original training data, the higher accuracy the classification results would get.

We compare the proposed algorithm, Smedoids, with several state-of-the-art methods for finding representatives: Kmedoids, Sparse Modeling Representative Selection (SMRS) and simple random selection (Rand). Two standard classification algorithms, multi-class classifier (SVM) [20] and Sparse Representation-based Classification (SRC) [23], are used to evaluate the multi-class classification performance. The experiments are run on the handwritten digits database USPS [21] and the Yale Face Database B [22]. The USPS handwritten database contains 11000 images of ten digit characters, and in each class 1000 samples are randomly selected for training and left for testing. The Yale-B contains 5760 images of 10 subjects which have been cropped to the size of 16 by 19 pixels by us, and in each class 300 samples are randomly selected for training and left for testing. We run several times with different number l of representatives selected from training set. Obviously with

more training representatives, the classification will achieve higher accuracy. Tables 1 and 2 show the classification results for the USPS database and the Yale-B database respectively.

From the results, we can conclude that our proposed method always gets the best accuracy. All representative selection methods work better with SVM than with SRC. In contrast to [9] SMRS performs better than Kmedoids in some cases but not always.

Table 1. Classification results on USPS digit dataset using l representatives of the 1000 training samples in each class

USPS Representatives #		Rand	Kmedoids	SMRS	Proposed
SRC	l =10	0.77	0.838	0.824	**0.86**
	l =20	0.86	0.872	0.868	**0.896**
	l =30	0.895	0.898	0.902	**0.92**
	l =40	0.9127	0.917	0.917	**0.928**
SVM	l =10	0.8027	0.8758	0.8556	**0.907**
	l =20	0.8809	0.888	0.8697	**0.9236**
	l =30	0.9137	0.9308	0.9243	**0.9464**
	l =40	0.9346	0.9392	0.9305	**0.9489**

Table 2. Classification results on Yale-B Face dataset using l representatives of the 300 training samples in each class

Yale-B Face Representatives #		Rand	Kmedoids	SMRS	Proposed
SRC	l =10	0.44	0.5117	0.4633	**0.54**
	l =20	0.57	0.6397	0.5947	**0.6877**
	l =30	0.6347	0.7067	0.6990	**0.74**
	l =40	0.7003	0.7627	0.754	**0.7843**
SVM	l =10	0.5873	0.6417	0.6017	**0.7083**
	l =20	0.7213	0.7877	0.7787	**0.834**
	l =30	0.7997	0.8277	0.8397	**0.868**
	l =40	0.8443	0.889	0.8787	**0.9003**

Our proposed method works best because not only center points but also boundary points are selected. We investigate the effect of the parameters of algorithm1, T (the number of iterations) and s (the maximal number of nonzero atoms in each sparse representation). We set T to 20 and s to 5 in all runs, and we also constraint the sparse representations to be non-negative to get the reported result.

5 Conclusion

In this paper, we propose a Smedoids algorithm to select representatives from entire dataset and prove its convergence. The Smedoids algorithm selects both center and boundary points that well cover the whole distribution of dataset, and we argue that our method can condense the original dataset better than the state-of-the-art representative selection methods. Results of video summarization show that main activities of each shot are well captured. In addition our proposed method always achieves the best accuracy among the state-of-art algorithms using representatives for multi-class classification problem.

Acknowledgement. This work was supported in part by the National Nature Science Foundation of China (61173054, 61271428); and Co-building Program of Beijing Municipal Education Commission.

References

1. Kaufman, L., Rousseeuw, P.: Clustering by means of medoids. In: Dodge, Y. (ed.) Statistical Data Analysis based on L1 Norm. North-Holland, Amsterdam (1987)
2. Jurie, F., Triggs, B.: Creating Efficient Codebooks for Visual Recognition. In: ICCV (2005)
3. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science (2007)
4. Boutsidis, C., Mahoney, M.W., Drineas, P.: An improved approximation algorithm for the column subset selection problem. In: Proc. SODA (2009)
5. Balzano, L., Nowak, R., Bajwa, W.: Column subset selection with missing data. In: NIPS Workshop on Low-Rank Methods for Large-Scale Machine Learning (2010)
6. Tang, S., Zheng, Y.-T., Wang, Y., Chua, T.-S.: Sparse Ensemble Learning for Concept Detection. IEEE Trans on Multimedia 14(1), 43–54 (2012)
7. Bien, J., Tibshirani, R.: Prototype selection for interpretable classification. The Annals of Applied Statistics (2011)
8. Marchiori, E.: Class conditional nearest neighbor for large margin instance selection. IEEE Trans. PAMI 32(2), 364–370 (2010)
9. Elhamifar, E., Sapiro, G., Vidal, R.: See all by looking at a few: sparse modeling for finding representative objects. In: CVPR (2012)
10. Aharon, M., Elad, M., Bruckstein, A.M.: The k-svd: An algorithm for designing of overcomplete dictionaries for sparse representations. IEEE Trans. SP 54(11), 4311–4322 (2006)
11. Ramirez, P., Sprechmann, G.: Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features. In: CVPR (2010)
12. Sprechmann, P., Sapiro, G.: Dictionary Learning and Sparse Coding for Unsupervised Clustering. In: ICASSP (2010)
13. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: ICML (2007)
14. Mairal, J., Bach, F., Ponce, J.: Task-driven dictionary learning. IEEE Trans. on PAMI 34(4), 791–804 (2011)
15. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research 11, 19–609 (2010)

16. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. on PAMI 31(2), 210–227 (2009)
17. Tibshirani, R.: Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society. Series B 58(1), 267–288 (1996)
18. Zhang, Y., Yan, C., Dai, F., Ma, Y.: Efficient Parallel Framework for H.264/AVC Deblocking Filter on Many-core Platform. IEEE Trans. on Multimedia 14(3), 510–524 (2012)
19. Vidal, R.: Recursive identification of switched ARX systems. Automachine (2008)
20. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 3(2), 1–27 (2011), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
21. Hull, J.: A database for handwritten text recognition research. IEEE TPAMI (1994)
22. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE TPAMI (2005)
23. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. on PAMI 31(2), 210–227 (2009)
24. Wang, M., Hong, R., Li, G., Zha, Z.-J., Yan, S., Chua, T.-S.: Event Driven Web Video Summarization by Tag Localization and Key-Shot Identification. IEEE Trans. on Multimedia 14(4), 975–985 (2012)
25. Hong, R., Wang, M., Xu, M., Yan, S., Chua, T.-S.: Dynamic Captioning: Video Accessibility Enhancement for Hearing Impairment. In: ACM MM (2010)