International Journal of Electrical, Electronics and Data Communication, ISSN(p): 2320-2084, ISSN(e): 2321-2950
Volume-7, Issue-10, Oct.-2019, http://iraj.in

IMAGE SELECTION USING NON-LINEAR SPARSE REPRESENTATION SCHEMES

[1]F. DORNAIKA, [2]A. ASSOUM, [3]I. KAMAL ALDINE

[1,3]Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastian, Spain
[1]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[2]Lebanese University, LaMA Laboratory, Tripoli, Lebanon
E-mail: fadi.dornaika@ehu.es, a.assoum@ul.edu.lb, ihab4@hotmail.com

Abstract - Sparse Modeling Representative Selection (SMRS) has been recently introduced for selecting the most relevant instances in datasets. SMRS utilizes data self-representativeness coding in order to infer a coding matrix with block sparsity constraint. The relevance scores of any instance is then set to the ℓ_2-norm of the corresponding row in the coding matrix. Since SMRS is based on a linear model for data self-representation, it cannot always provide good relevant representative instances. Besides, most of its selected instances can be found in dense areas in the input space. In this chapter, we propose to overcome the SMRS method's shortcomings that are related to the coding matrix estimation. We introduce two non-linear data self-representativeness coding schemes that are based on Hilbert space and column generation. Experimental evaluation is carried out on summarizing a video movie and on summarizing training image datasets used for classification tasks. These experiments demonstrated that the proposed non-linear methods can outperform state-of-the-art selection methods including the SMRS method.

Keywords - Block Sparsity, Column Generation, Data Self-Representativeness, Hilbert Space, Instance Selection, Kernel Representation.

I. INTRODUCTION

Estimating a subset of instances, known as representatives, that can efficiently and reliably describe the whole dataset, is an important issue in the analysis of scientific data. It has a lot of applications in machine learning, data recovery, image processing, etc. Due to the effectiveness of instance selection for speeding up training processes, many methods have been proposed [2]-[15]-[20]. The selected representatives can summarize datasets of images, videos, texts or Web documents. Finding a small number of in- stances which replaces the learning database has two main advantages: (i) reducing the memory space needed to store data, and (ii) improving the computation time of classification algorithms. For example, the Nearest Neighbor (NN) classifier is more efficient [11] when comparing test samples to few representatives rather than to all training samples. A reduced training dataset can also speed up the training process in the sense that the classifier learning becomes less computationally expensive. For pattern recognition tasks, it is also required that the overall performance will not be considerably affected by the data reduction. The problem can be stated as follows: given a training set T, the goal of an instance selection method is to obtain a subset $S \subseteq T$ such that S does not contain superfluous instances and $Acc(S) \approx Acc(T)$ where $Acc(S)$ is the classification accuracy obtained using the subset S as training set. Instance selection methods can either start with $S = \phi$ (incremental methods) or $S = T$ (decremental methods).
Like in feature selection, according to the strategy used for selecting instances, we can divide the instance selection methods in two groups: (i) Wrapper methods in which the selection criterion is based on the accuracy obtained by a classifier (commonly, those instances that do not contribute with the classification accuracy are discarded from the training set) (e.g. [7]-[5]), and (ii) Filter methods in which the selection criterion uses a selection function which is not based on a classifier (e.g. [16]). A good review on wrapper and filter methods can be found in [18]. Most of the instance selection algorithms (e.g. [6]) are strongly related to the use of the k-NN classifier. One can also find instance selection algorithms that do not restrict the use of a specific classifier.
The filter algorithms can be divided into two main groups. The first category finds representatives from data contained in one or several subspaces of reduced dimensionality. For instance, the algorithm *Rank Revealing QR (RRQR)* [3] tries to select a few data points through finding a permutation of the data which gives the best conditioned submatrix. *Greedy* and *Randomized* algorithms have also been proposed in order to find a subset of columns in a reduced rank matrix [2].

The second group of algorithms finds representatives assuming there is a natural grouping of data collection based on an appropriate measure of similarity between pairs of data points [13]-[10]-[12]. Accordingly, these algorithms generally work on the similarity/dissimilarity between data points to be grouped. The *Kmedoids* algorithm [13], which can be considered as a variant of *Kmeans* [8], supposes that the data are located around several centers of classes, called medoids, which are selected from the data. Another algorithm based on the similarity/dissimilarity of data points is the *Affinity propagation (AP)* [12].

A. Motivation and contribution

Recently, a new filter method, called *Sparse Modeling Representative Selection (SMRS)* [9], has been proposed to find sample representatives and is based on setting every data sample as a linear combination of the whole dataset with a block-sparsity constraint. SMRS is essentially based on the assumption that for each sample in the dataset there exist some samples that form a linear subspace that the sample belong to or very close to it.

This assumption is very similar to the one used by the Locally Linear Embedding (LLE) technique where each sample is assumed to be an affine combination of neighboring samples [19]. In SMRS method, the whole dataset is used as a dictionary and the block sparsity is imposed on the matrix of coding coefficients in order to enhance the relevance of samples. SMRS suffers from at least two shortcomings. First, the linear assumption can be violated. Indeed, real-world data usually have non-linear distributions such that, even in local neighborhoods, a linear subspace can be a rough approximation. Second, due to the use of a linear model for a relatively big dictionary, samples belonging to dense regions in data space will have large coefficients. As a consequence, SMRS tends to select the majority of relevant instances in dense regions. This can be undesirable from the point view of classification where the presence of samples at class borders can enhance the discrimination between different classes.

In this paper, we propose to overcome these shortcomings. We will use kernel sparse subspace modeling where the linear combination is performed in high dimension space in the hope that sample relevance can be better captured in these high dimensional spaces. In the literature, the kernel trick has been used to get the non-linear variant of many linear embedding techniques such as Linear Discriminant Analysis (LDA) and Local Discriminant Embedding (LDE).

The paper is structured as follows: in Section 2, we review the SMRS method. In Section 3, we describe our proposed kernel methods. Section 4 presents a qualitative evaluation on video summarization and a quantitative evaluation that quantifies the classification performance based on the selected instances. Finally, we provide some concluding remarks in Section 5. In the sequel, capital bold letters denote matrices and small bold letters denote vectors.

II. REVIEW OF SPARSE MODELING REPRESENTATIVE SELECTION

In this section, we briefly describe the Sparse Modeling Representative Selection (SMRS) method proposed in [9]. The problem formulation can be stated as follows. Consider a set of data samples $T = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ in $\mathbb{R}^d$ arranged as the columns of the data matrix $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]$, where d denotes the sample dimension. The objective is to select the most representative samples within the set of samples T. SMRS is a filter method that uses the concept of relevance ranking. In other words, the relevance score of every sample $\mathbf{y_i}, i = 1, \dots, N$ is first estimated. Based on the sorted relevance scores, the most relevant instances are then selected using a pre-defined threshold or a fixed number of instances. The basic idea of [9] is to estimate the data self-representativeness coefficients from which the relevance score of each sample can be derived. The basic assumption is that every data sample is equal or close to a linear combination of some samples in the original dataset. The unknown coding matrix can be estimated by minimizing the following criterion:

$$\mathbf{B} = \arg\min_{\mathbf{B}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{YB}\|_2^2 + \lambda \|\mathbf{B}\|_{1,2} \right) s.t.\ \mathbf{1}^T\mathbf{B} = \mathbf{1}^T$$

where λ is a positive regularization parameter. The above criterion has two terms. The first is the least square error associated with the self-representativeness of data. The second is the $L_{1,2}$ norm of the matrix $\mathbf{B}$, i.e. $\|\mathbf{B}\|_{1,2} = \sum_j \left[\left| b_{(j,1)} \right|^2 + \cdots + \left| b_{(j,N)} \right|^2 \right]^{\frac{1}{2}}$ (sum of L_2 norm of the rows of the matrix $\mathbf{B}$). The framework for this selection is summarized in Algorithm 1. In a supervised context, it is applied on each class separately in order to retrieve the most representative samples in each class.

Data: A dataset $\mathbf{Y} \in \mathbb{R}^{d \times N}$, a regularization parameter λ, a threshold $th \in [0.9, 0.99]$ for selecting the non-zero rows (the representatives)
Result: Set of representatives $\mathbf{Y}_s$

$\mathbf{Y}_s = \emptyset$;
For each class $c = 1, \dots, C$ do;
- Calculate the coding coefficients $\mathbf{B}_c$ associated with the samples of class c using $\mathbf{B}_c = \arg\min_{\mathbf{B}_c} \left(\frac{1}{2} \|\mathbf{Y}_c - \mathbf{Y}_c \mathbf{B}_c\|^2 + \lambda \|\mathbf{B}_c\|_{1,2} \right) s.t.\ \mathbf{1}^T \mathbf{B}_c = \mathbf{1}^T$ where $\mathbf{Y}_c$ is the dataset associated with class c;
- Compute the L_2 norms of the rows of $\mathbf{B}_c$ as $l_i = \|\mathbf{B}_c(i,:)\|, i = 1, \dots, N_c$;
- Sort the scores $l_i, i = 1, \dots, N_c$ in a decreasing order;
- Calculate the smallest integer K such that $\sum_{j=1}^{K} l_j / \sum_{j=1}^{N_c} l_j > th$;
- Select the first K examples in the ordered list $\{\mathbf{y}_1^*, \dots, \mathbf{y}_K^*\}$;
- $\mathbf{Y}_s = \mathbf{Y}_s \cup \{\mathbf{y}_1^*, \dots, \mathbf{y}_K^*\}$;

End

Algorithm 1: Selecting class representatives using SMRS.

III. PROPOSED KERNEL SPARSE MODELING

In this section, we introduce two kernelized sparse modeling selection schemes. The first one adopts the projection onto Hilbert space. The second uses the trick of column generation. In fact, in several important problems in computer vision such as face recognition and activity recognition, the data can be well approximated by a union of subspaces. These subspaces are observed in the ambient space in which data distribution can have high non-linearity.

A. Hilbert space

The motivation behind the use of kernel representation relies on the fact that a linear model for data self-representativeness is not the best model for estimating the matrix of coefficients from which the instances are selected. Thus, by adopting non-linear

models for data self-representativeness, it is expected that the resulting coding coefficients could better quantify the dependency and relation among samples and hence, better coefficients and selection can be obtained whenever data have non-linear distribution that can be observed even in local neighborhoods. Fig. 1 sketches out the configuration of data distribution in original and Hilbert spaces.

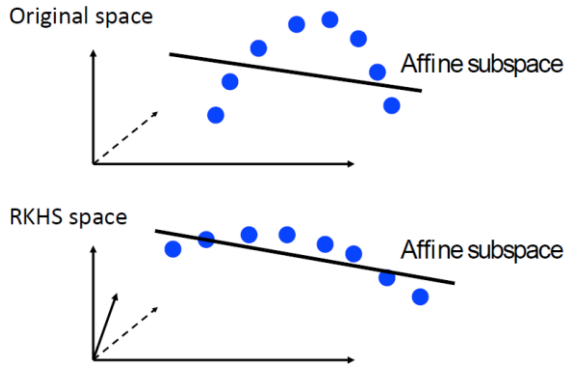

Figure 1: Top: Possible non suitability of affine subspace for data self-representation in original spaces. Bottom: Suitable affine subspace in projected Hilbert spaces.

Let $\Phi : \mathbf{Y} \rightarrow \Phi(\mathbf{Y})$ be a non-linear projection that maps original samples onto a space of high dimension. According to the kernel theory, it is not necessary to know the explicit mapping Φ since what is really needed is the dot product among the obtained projections. In the new space, the data are represented by the matrix $\Phi = [\phi(y_1), \phi(y_2), \ldots, \phi(y_n)]$. Let $K_{ij} = \phi^T(y_i)\phi(y_j)$ be the dot product of the projection of two samples y_i and y_j . This dot product introduces a kind of similarity measure between samples y_i and y_j . The kernel matrix $K(.,.)$ can be built using Gaussian, polynomial, or any other function that satisfy Mecer's conditions. It is easy to show that the matrix $\mathbf{K}$ will be given by $\Phi^T\Phi$.

By adopting the projected data, Φ, the proposed kernel method can be obtained by replacing the data with their non-linear projection. Thus, we have:

$$\mathbf{B} = \arg\min_{\mathbf{B}} \left(\frac{1}{2}\|\Phi - \Phi\mathbf{B}\|_2^2 + \lambda\|\mathbf{B}\|_{1,2}\right) s.t.\ \mathbf{1}^T\mathbf{B} = \mathbf{1}^T \quad (1)$$

The affinity constraint $\mathbf{1}^T\mathbf{B} = \mathbf{1}^T$ can be augmented to the linear system $\Phi = \Phi\mathbf{B}$. In practice, we found that a simple normalization of the columns of the obtained matrix can also be used without deteriorating the global performance. Thus, our criterion to be minimized becomes:

$$\mathbf{B} = \arg\min_{\mathbf{B}} \left(\frac{1}{2}\|\Phi - \Phi\mathbf{B}\|_2^2 + \lambda\|\mathbf{B}\|_{1,2}\right) \quad (2)$$

The above optimization problem can be solved using the Alternating Direction Method of Multipliers (ADMM) [4] which allows us to employ a dummy variable into the objective such that:

$$\arg\min_{\mathbf{B}} \left(\frac{1}{2}\|\Phi - \Phi\mathbf{B}\|_2^2 + \lambda\|\mathbf{C}\|_{1,2} + \frac{\rho}{2}\|\mathbf{B} - \mathbf{C}\|^2\right) s.t.\ \boldsymbol{B} = \boldsymbol{C}$$

The Lagrangian of the above equation can be written as:

$$L(\mathbf{B}, \mathbf{C}) = \frac{1}{2}\|\Phi - \Phi\mathbf{B}\|_2^2 + \lambda\|\mathbf{C}\|_{1,2} + \frac{\rho}{2}\|\mathbf{B} - \mathbf{C}\|^2 + Tr\left(\Lambda^T(\mathbf{B} - \mathbf{C})\right)$$

where $Tr(.)$ denotes the matrix trace and Λ the Lagrangian multiplier matrix. The ADMM optimization is described in Algorithm 2. Once the sparse coefficients are estimated, the selection will be based on the sorted l_2-norms of the rows.

Input: The data matrix $\mathbf{Y}$, the regularization parameter λ, ϵ (small positive threshold), ρ, $IteMax$
Output: The matrix of coefficients $\mathbf{B}$

Set the kernel matrix $\mathbf{K}$, where $K_{ij} = sim(\mathbf{x}_i, \mathbf{x}_j)$;
Initialize $\mathbf{C}_0$ to a random matrix;
Initialize Λ_0 to zero matrix;
t=0 ;
repeat
1. $\mathbf{B}_{t+1} = (\mathbf{K} + \rho\mathbf{I})^{-1}(\mathbf{K} + \rho\,\mathbf{C}_t - \Lambda_t)$;
2. $\mathbf{C}_{t+1} = SoftThresholding(\mathbf{B}_{t+1} + \Lambda_t/\rho, \lambda/\rho)$;
3. $\Lambda_{t+1} = \Lambda_t + \rho\,(\mathbf{B}_{t+1} - \mathbf{C}_{t+1})$;
$t = t + 1$;

until $\|\mathbf{B}_{t-1} - \mathbf{C}_{t-1}\|_2^2/N^2 < \epsilon$ *or* $t > IteMax$;

Algorithm 2: ADMM optimization.

B. Column generation

Column generation replaces each sample y_i by a vector of similarities of that sample with the samples contained in a fixed set of samples [14]. Very often, the latter set is given by the training samples. The data matrix $\mathbf{Y}$ is thus replaced by the matrix $G = [g_1, g_2, \ldots, g_n]$, where each vector g_i is formed by the similarities, i.e. $g_i = [sim(x_i, x_1), \ldots, sim(x_i, x_n)]^T$. The optimization problem becomes:

$$\mathbf{B} = \arg\min_{\mathbf{B}} \left(\frac{1}{2}\|\mathbf{G} - \mathbf{GB}\|_2^2 + \lambda\|\mathbf{B}\|_{1,2}\right) \quad (3)$$

The solution to (3) is again similar to Algorithm 2, replacing $\mathbf{G}^T\mathbf{G}$ by $\mathbf{K}$.

IV. PERFORMANCE EVALUATION

In this section, we first provide a qualitative evaluation of the SMRS method and the proposed method when applied to video sequence summarization. We then provide classification results over two benchmark image datasets after selecting training representatives using different competing selection methods and different classifiers.

A. Video summarization

We consider a 4148-frame video retrieved from YouTube [1]. This video is called "Raffles (Preview

[1] https://www.youtube.com/watch?v=ZqaYGxroOis

Clip)". It consists of a series of continuous activities with a variable background. We apply the SMRS method and the proposed kernel method in order to obtain 35 representative images among the 4148 images. Then, these 35 representatives are pruned such that very close representatives (in time) are merged into one single frame. Figures 2(a) and 2(b) show the final image representatives obtained by the SMRS method and the proposed kernel method, respectively. Note that the representatives obtained by the algorithms captured the main events and scenes of the video. We can observe that the representative images selected by the SMRS method and the kernel method are not the same. Furthermore, we can see that the SMRS method obtained more close representatives than the proposed one. We can easily see that the SMRS method selected five representative images that belong to almost the same activity (video shot) (see selected frames 8, 9, 10, 11, and 12). On the other hand, the kernel method has preferred to get representatives images from the whole video. This observation can be explained by the fact that the SMRS method tends to select representatives in dense regions.

(a) Representatives from the 4148 images using the SMRS method.

(b) Representatives from the 4148 images using the proposed kernel method.

Figure 2: Summarizing the video "Raffles (Preview Clip)". The automatically computed representatives using the SMRS method (a), and the proposed kernel method (b). Depending on the amount of activities in each shot of the video and on the method used, we obtained one or a few representatives for that shot. We can observe that, unlike the SMRS method, the kernel method has evenly selected frames from the whole video.

B. Pattern classification using selected instances

Datasets: In our experiments, we considered two public image datasets, which are characterized by large variation in appearance. We used a digit dataset and a face dataset.

USPS[2]: Handwritten Digit dataset composed of 11000 grayscale images of ten digits "0" through "9"; each class (digit) has 1100 images.

Extended Yale - part B[3]: It contains 38 human subjects each has about 60 images.

Evaluation protocol: We evaluate the performance of our method and other algorithms for selecting instances that are used for classification. For the quantitative evaluation, we compare the proposed method with the following instance selection methods: Kmedoids, Collaborative Neighborhood Representation (CNR) [21] (this is a constrained coding scheme), simple random selection of training data (Rand), PSR [17], DROP3 [22], and SMRS [9]. The performance is quantified by the classification accuracy (recognition rate) using the obtained instances as training data. We adopt a commonly used protocol in order to evaluate the classification accuracy after instance selection. First, every dataset is randomly split into two parts: a training part and a test part. Then the training part is processed by the instance selection method that selects a set of training

[2] http://www.cs.nyu.edu/~roweis/data.html
[3] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

representatives. The test part is then recognized using either all training data (in the tables depicting the recognition results, this is referred as to "All data") or the selected representative training data (in the tables, each method corresponds to a row). This automatic recognition is carried out by three classifiers: Nearest Neighbor (NN), Sparse Representation Classifier (SRC) [23], and Support Vector Machines (SVM). Then, given the ground-truth labels of the test set, the recognition performance based on each instance selection method can be quantified. Since the same training data and test data are used, the difference in classification performance will be certainly due to the selection algorithm alone. For fair comparison, all selection algorithms are used such a way they provide the same number of selected instances. The kernel function used for the two proposed schemes is the Gaussian kernel. Tables 1 and 2 illustrate the recognition performance on the two datasets. The last row in each table illustrates the performance using the whole training samples. In these tables, we used a fixed number of representatives per class. The DROP3 method is a wrapper method that does not have any parameter that specifies the number of selected examples. In order to get a fair comparison with the rest of methods we clapped the number of selected samples by DROP3 by the number used by all methods. Finally, for every dataset, the number of selected instances was kept fixed for all compared methods.
Table 1 illustrates the classification performance on the USPS dataset after the selection of 25 representatives of the 1000 training samples in each class. The number of test images is 100 images per class. Table 2 illustrates the recognition performance on the Extended Yale face dataset after the selection of 12 representatives of the 48 training samples in each class. The number of test images is 10 images per class. For this dataset, we use the Local Binary Patterns images (LBP) [1].

Analysis of results: From the above results, we can observe the following (1) The proposed kernel methods (Kernel Sparse Modeling and CG Sparse Modeling) outperformed the competing selection methods including the SMRS method. (2) There is no significant difference in the performances of the two proposed methods. Furthermore, there is no final conclusion on which method is the best. (3) The difference between the performance obtained with the selected representatives (using the two proposed methods) and that obtained with the whole dataset is small. For example, when the proposed method is used with the SRC classifier, these differences are 2.4% and 0.8% for USPS dataset and Extended Yale dataset, respectively.

Table 3 illustrates the recognition performance obtained with the proposed (Kernel Sparse Modeling method) on two datasets for different kernel types. As can be seen, the Gaussian kernel provides the best results for the tested datasets.

Table 1: Classification performance (%) after instance selection (USPS dataset).

Selection method \ Classifier	NN	SRC	SVM
Rand	88.6	93.0	92.4
K-medoids	88.8	92.6	93.0
PSR	82.8	85.4	80.0
DROP3	87.0	91.2	89.4
CNR	80.9	86.1	90.0
SMRS	81.6	90.4	95.2
Kernel Sparse Modeling	**93.0**	**95.6**	**96.8**
CG Sparse Modeling	**91.4**	**95.4**	**97.2**
All data	97.2	98.0	98.8

Table 2: Classification performance (%) after instance selection (Extended Yale dataset).

Selection method \ Classifier	NN	SRC	SVM
Rand	94.5	97.9	97.7
K-medoids	96.7	98.9	98.1
PSR	93.1	96.5	89.8
DROP3	96.3	97.1	97.1
CNR	94.3	96.1	93.7
SMRS	93.9	97.7	97.7
Kernel Sparse Modeling	96.1	**99.2**	**98.8**
CG Sparse Modeling	**96.9**	**99.2**	**99.3**
All data	100	100	99.5

Table 3: Classification performance (%) after instance selection.

USPS: Kernel type \ Classifier	NN	SRC	SVM
Polynomial d=1	65.2	80.2	85.6
Polynomial d=2	58.0	71.6	77.0
Polynomial d=3	58.0	71.6	77.0
Gaussian	**93.0**	**95.6**	**96.8**
Ext. Yale: Kernel type \ Classifier	NN	SRC	SVM
Polynomial d=1	91.7	96.8	96.1
Polynomial d=2	93.3	97.5	96.9
Polynomial d=3	93.7	98.5	97.3
Gaussian	**96.1**	**99.2**	**98.8**

V. DISCUSSIONS AND CONCLUSIONS

We proposed novel instance selection methods for finding representatives in a given set of data samples. The proposal methods were based on non-linear Sparse Modeling Representation. We compared our proposed methods with several competing methods as well as with the recent SMRS method. Experimental results on public databases and a video movie are presented to demonstrate the efficacy of the proposed approaches. The databases correspond to images, which makes the selection and classification more challenging. The classification results, conducted with three different classifiers, have shown that the proposed methods can provide better results than many state-of-the art instance selection methods. Compared with the SMRS method, the proposed method has overcome the non-linearity data representation due to the implicit

projection onto Hilbert space or to the use of the Column Generation based projection.

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[2] J. Bien, Y. Xu, and M.W. Mahoney. CUR from a sparse optimization viewpoint. In *Advances in Neural Information Processing Systems*, pages 217–225, December 2010.

[3] C. Boutsidis, M.W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 968–977, January 2009.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[5] J. Chen, C. Zhang, X. Xue, and C-L. Liu. Fast instance selection for speeding up support vector machines. *Knowledge-Based Systems*, 47:1–7, 2013.

[6] C. Chien-Hsing, K. Bo-Han, and C. Fu. The generalized condensed nearest neighbor rule as a data reduction method. In *IEEE International Conference on Pattern Recognition*, 2006.

[7] I. Czarnowski. Cluster-based instance selection for machine classification. *Knowledge and Information Systems*, 78(3):1–21, 2010.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley Inter- science, 2004.

[9] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1600–1607, June 2012.

[10] B.J. Frey and D. Dueck. Clustering by passing messages between data points. *Science Magazine*, 315:972–976, 2007.

[11] S. Garcia, J. Derrac, R. Cano, and F. Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[12] I.E. Givoni, C. Chung, and B.J. Frey. Hierarchical affinity propagation. In *Conference on Uncertainty in Artificial Intelligence*, July 2011.

[13] L. Kaufman and P. Rousseeuw. *Statistical Data Analysis based on the L1-Norm*, chapter Clustering by means of medoids, pages 405–416. 1987.

[14] B. Klare and A. Jain. Heterogeneous face recognition using kernel prototype similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):1410–1422, 2013.

[15] Y. Li and L. Maguire. Selecting critical patterns based on local geometrical and statistical information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1189–1201, 2011.

[16] B.L. Narayan, C.A. Murthy, and S.K. Pal. Maxdiff kd-trees for data condensation. *Pattern Recognition Letters*, 27:187–200, 2006.

[17] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, and J. F. Martinez-Trinidad. Prototype selection via prototype relevance. In *IberoAmerican Congress on Pattern Recognition*, LNCS 5197, 2008.

[18] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, J. F. Martinez-Trinidad, and J. Kittler. A review of instance selection methods. *Artificial Intelligence Review*, 34:133–143, 2010.

[19] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[20] J.A. Tropp. Column subset selection, matrix factorization and eigenvalue optimization. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 978–986, January 2009.

[21] J. Waqas, Z. Yi, and L. Zhang. Collaborative neighbor representation based classification using l_2-minimization approach. *Pattern Recognition Letters*, 34(2):201–208, 2013.

[22] D.R. Wilson and T.R. Martinez. Reduction techniques for instance- based learning algorithms. *Machine Learning*, 38:257–286, 2000.

[23] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

★★★